www.ijecs.in
International Journal Of Engineering And Computer Science ISSN: 2319-7242
Volume 5 Issue 11 Nov. 2016, Page No. 19045-19049

Efficient Exploration of Algorithms in Scholarly Documents Using Big Data Analytics

B. Tamil Bharathi [1], K. Devaki[2]
[1] PG student, Department of Computer Science and Engineering,
Rajalakshmi Engineering College,
Thandalam, Chennai, Tamil Nadu.
tamilbharathi94@gmail.com
[2] Professor, Department of Computer Science and Engineering,
Rajalakshmi Engineering College,
Thandalam, Chennai, Tamil Nadu.
devaki.k@rajalakshmi.edu.in

ABSTRACT: ***Algorithms are important and essential part of computational science research work which is regularly published in scholarly articles. To solve the scaling problem of handling more documents, we examine an intelligent system designed with the goal of automatically identifying algorithms. In this paper, we propose a method to develop an algorithm search engine. The proposed system analyzes a document to discover any algorithm that may be there in the document. If any algorithm is found in the document, the document text is further analyzed to extract additional information about the algorithm. Machine learning and rule based approaches are used to discover algorithm representation. All discovered algorithms and their associated metadata are indexed and offered for searching throughout a text query interface.***

Keywords: Algorithm search, indexing, rule based, machine learning.

1. Introduction

Scholarly big data indicates the large amount of data that is related to scholarly documents, such as journal articles, conference proceedings, books, patents, theses and experimental data. Big data is explained by a set of V's, which initially referred to volume, velocity and variety but also include other concepts such as value, vulnerability, veracity and viscosity. As facts of the volume and velocity of scholarly big data, Microsoft Academic is reported to contain above 50 million academic document records and in 2010 it was estimated that the annual growth rates of various popular databases from 1997-2006 range from 2.7 to 13.5%. Moreover, a large amount of that data is liberally available on the Web with a modern study finding that an average of 43% of articles published between 2008 and 2011 were freely accessible online. A significant challenge for these services is to deal with the scale of the data they collect, integrate information from several sources and extracting meaningful information from the data.

Researchers are improving the advanced algorithms and developing new algorithms for new and unsolved problems; frequently researchers report their new algorithms in scientific publication. It would be useful to have automated systems that efficiently identify, extract, index, and search. There is still increasing collection of algorithm modernization and such systems might provide another source for researcher and software developer looking for advanced algorithmic solutions to their problems. Discovering well know standard algorithms is not difficult, as they were

previously collected and cataloged manually in algorithm text book (e.g. encyclopedias) and made searchable, especially those in online catalogs.

Recently published algorithms are a nontrivial task for automatic searching process. Researchers and others are aiming to find out efficient and new algorithms in their field to keep abreast of the latest algorithmic developments. Having to read an entire document is tedious. The problem is worse for algorithm searchers who are inexperienced in document search. Perfectly, we would like to have a system that automatically discovers and extracts algorithms from scholarly digital documents. Such a system automatically identifies and extracts algorithms; in particular their pseudo codes since many algorithms are read as such from digital documents.

2. Background and Related Works

2.1 Document Entities Search

Finding and extracting informative entities such as mathematical expressions, tables, figures and tables of contents from documents have been studied. Bhatia et al., describe methods for automatic detection of algorithm in [2] Computer Science documents. Their system considers that each algorithm is accompanied by a caption. Such an algorithm can then be identified using a set of regular expressions to detect the presence of the accompanied caption. Such an approach, however, is limited in its coverage due to its reliance on the presence of algorithm captions and wide variations in writing styles and some scholarly documents which did not have an accompanied caption. Thus, these algorithms will remain undetected by their approach. Since algorithms representation in documents do not be conventional to specific styles, and are written in arbitrary formats, this becomes a challenge for effective detection and extraction.

Kataria, et al.[7], employed image processing and Optical Character Recognition (OCR) approaches for extraction of data points and text blocks from 2D plots. They also proposed a system to index and search for the extracted information. Sojka proposed MIaS (Math Indexer and Searcher) that collects and interprets mathematical expressions. Their system

only handles documents in the MathML (Mathematical Markup Language) format where mathematical expressions have already been marked up.

M. Khabsa et al.,[6] had demonstrated AckSeer, a search engine and a repository for automatically extracting acknowledgments in digital libraries. AckSeer is a fully automated system that scans objects in digital libraries including conference papers, journals, and books extracting acknowledgment sections and identifying acknowledged entities mentioned within. In the section, extraction of algorithm is based on regular expressions, but it is robust enough with average F1 score 85%. They used multiple Named Entity Recognition (NER) tools and designed a method for integrating the outcome from dissimilar recognizers. The resulting entity are stored in a database and then made searchable by adding them to the AckSeer index along with the metadata of the containing paper/book.

2.2 Search Systems for Scholarly Information

There are various search engines that have been proposed for document search. CiteSeerX[8], automatically crawls and indexes scientific documents, mostly in the field of computer and information science. Liu, et al.[9] presented TableSeer, that automatically identifies and extracts tables in digital documents. They used a tailored vector-space model based ranking algorithm, TableRank, to rank the search results. An implementation of TableSeer that extracts and searches for tables in the CiteSeerX document repository has been included in the CiteseerX suite. BioText8 search engine, a specialized search engine for biology documents, also have the capability to extract figures and tables, and make them searchable. Khabsa, et al.[6] described AckSeer, an acknowledgement search engine that automatically extracts, disambiguates, and indexes more than 4 million mentioned entities from 500,000 acknowledgments from documents in CiteSeerX[8]. CollabSeer[4], which is a search engine for discovering potential collaborators for a given author or researcher by analyzing the structure of the co-author network and the user's research interests.

TABLE 1 Author and their proposed search engine

AUTHOR	*PROPOSED SEARCH ENGINE*	*DOCUMENT ELEMENT*
h.Li, et all.,	*CiteSeerX*	*Crawls and Indexes*
Liu, et all.,	*TableSeer*	*Identifies and Extracts the Tables*
khabsa, et all.,	*AckSeer*	*Extracts Disambiguates and Indexes the acknowledgment*
Chen, et all.,	*CollabSeer*	*Discover collaborators for given author*

Here we have proposed a method for algorithm detection by capturing both algorithms with and without captions to overcome the challenges in the previous work.

3. System Architecture

Architecture of the proposed system is illustrated in Figure 1 we describe an algorithm search system using scientific publications as a document collection. Our proposed system analyzes a document to identify and extract the algorithm. Scholarly documents are processed to identify the algorithms and make them searchable. We now describe the different modules of the proposed system in more detail.

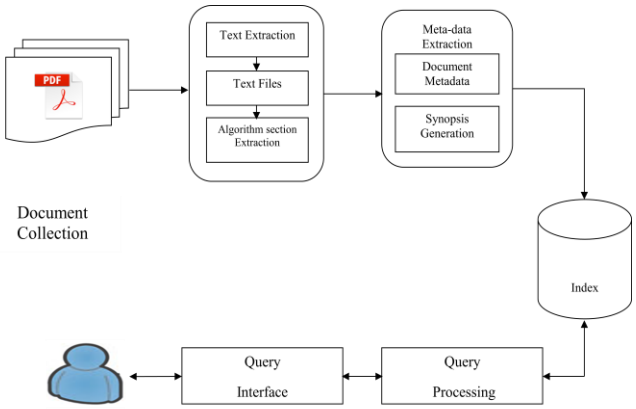


Figure 1 ARCHITECTURE DIAGRAM

3.1 Text Extraction

The source documents are taken from a scientific literature digital library. All documents in the collections are in PDF format and therefore need to be converted into text format for any further analysis. The document text is then analyzed to ensure presence of an algorithm. The extracted text is analyzed to find algorithms which are then indexed along with their associated meta-data.

3.2 Algorithm Identification

After document text has been extracted it needs to be analyzed to ensure if algorithm is present in a document. Scientific documents have a well defined structure. Regular algorithms/pseudocodes are described in the form of a standalone Figure or Table along with an associated caption and algorithm number by using these structural properties of scientific documents. We recognize whether the algorithm is present in a document. The algorithm caption and algorithm number is then used to refer to the algorithm in the text of the document and some algorithm did not have accompanied captions. Fig 2 represents example algorithm with caption and Fig 3 represents example algorithm without caption.

3.3 Meta-Data Extraction

If an algorithm is present, the document text is then further processed to extract the associated synopsis. Synopsis is a set of sentences that describe the algorithm for additional details on algorithm identification and synopsis generation. Reference sentence can be defined as a sentence which is referred to the algorithm. We can use these reference sentences to check for the presence of the algorithms in a document. For the documents containing an algorithm, furthermore we take out

the document title, author names, publication year and page on which the algorithm is present. For each extracted algorithms from a document and their associated meta-data are then indexed.

3.4 Query Interface

The proposed system offers a free text based query interface to the user and the results are returned along with the meta-data. The query processing system accepts the query from the user through the query interface, searches the index for related algorithms.

Algorithm OptimalHistogram()
Compute $\text{SUM}[1, i]$ *and* $\text{SQSUM}[1, i]$ *for all* $1 \leq i \leq n$
Initialize $\text{HERROR}[j, 1] = \text{SQSUM}[j, n], 1 \leq j \leq n$
1. *For j=1 to n do*
2. *For k=2 to B do*
3. *For i=1 to j-1 do*
4. $\text{HERROR}[j, k] = \min(\text{HERROR}[j, k], \text{HERROR}[i, k-1] + \text{SQERROR}[i+1, j])$

Figure 2. Algorithm OptimalHistogram

Fig. 2 Example Algorithm with caption

Initialization :
V = {}
E = {}
G = <V,E>

Begin :
for *each document* d *in* D:
N ← *list of algorithm-proposing documents cited in* d
For each (a, b) *where* a, b∈V, a ≠ b:
if edge (a, b) ∈E:
Increase weight of edge (a, b) *by* 1
Else :
Add edge (a, b) *to* E, *and set the weight to* 1
End If
Rerurn G
End.

Fig. 3 Example Algorithm without caption

4. Algorithm

Here we use two methods to identify algorithms in scholarly documents: Rule based, machine learning based and combined methods.

4.1 Rule Based Method:

Bhatia et al.,[1] proposed a rule based pseudo-code detection method, which utilizes a grammar for document-element captions to detect the presence of PC captions. In our method we have extended the previous work by adding rules to improve the effectiveness of algorithm detection.

Characteristics of Pseudocode:

- Named variables represent data and identifiers denote higher level functions.
- Composed of a sequence of statements or steps.
- Statements are frequently numbered sequentially.
- Operational (Imperative) statements include assignment, input, and output.
- Control structures provide iterative and conditional execution.
- Indentations used for grouping blocks of statements.

Here we extend the previous approach by adding the following rules to improve the coverage and reduce false positives:

- A PC caption should contain at least one algorithm keyword, namely pseudo-code, algorithm, and procedure.
- Captions in which the algorithm keywords appear after prepositions (e.g., 'Figure 5: Architecture proposed algorithm') are excluded, as these are not likely captions of PCs.
- Identify by keywords such as number of words, number of lines.
- Indentify by using alpha numeric characters, alphabets, symbols
- Identifying whether it contain stepwise indication word (example steps, following).
- Number of words in the paragraph is lesser than other paragraph.

Hence, given a document, the Rule Based method outputs a set of line numbers, each of which represents a pseudo-code caption.

4.2 Machine Learning Based Method:

In our dataset DS2, some of the pseudocodes do not have captions. These pseudocodes would remain undetected individually by the Rule Based method. To search documents, we use machine learning based method to identify the presence of pseudocode content instead of their captions.

This originates from the perception that most pseudocodes are written in a sparse method, resulting in sparse regions in the documents. This is known as sparse region's sparse boxes. The ML method first detects and extracts the sparse boxes present within the documents.

The subsequent subsections give details about the sparse box identification, the feature sets used

1) Sparse Box Extraction: A sparse box has a set of at least N consecutive sparse lines. Figure 4 shows an example of sparse boxes. A sparse line is a line whose ratio of the number of non-space characters to the average number of characters using line is less than threshold M.

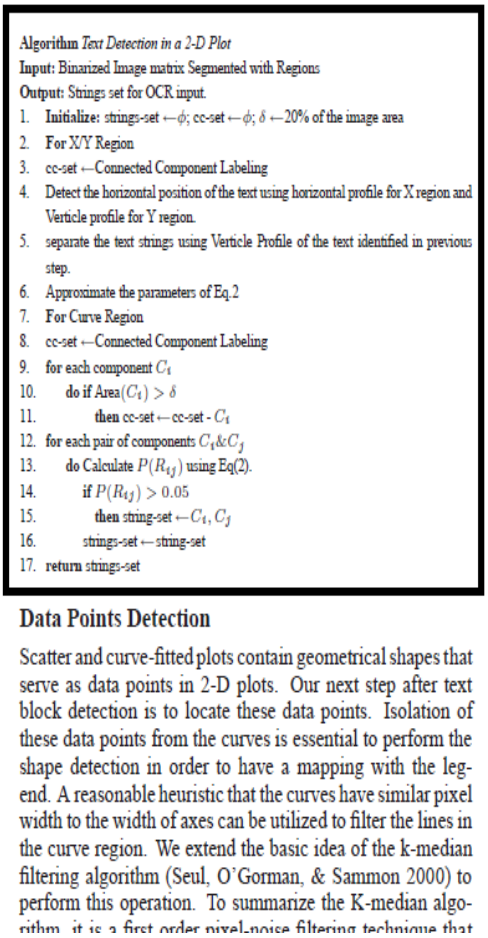
2) Feature Sets: Extract some features from each of the sparse boxes. These features are segregated into 4 groups: fontstyle based (FS), context based (CX), content based (CN), and structure based (ST). The Feature Set features capture the various font styles used in pseudocodes. The CX features capture the presence of pseudocode captions. The CN features detect the pseudocode by using exact keywords and coding styles. The ST features characterize the sparsity of pseudocodes and the symbols used. We adopt previous feature set and extend the feature set by adding Special character (SC). The SC features capture the presence of special characters in pseudocode.

5. Evaluation And Result

DOI: 10.18535/ijecs/v5i11.58

5.1 Datasets

We use two datasets; the first dataset (DS1) contains 100 scholarly documents manually selected from the scholarly digital library to cover dissimilar types of algorithms. This dataset is used to extract rules for rule-based methods and feature sets for proposed machine-learning based methods. The other dataset (DS2) consists of 250 scholarly PDF documents randomly selected from scholarly digital library consisting of 300 Algorithms used in validation. Fig.5 Screenshots shows the PDF to TEXT extraction.



Algorithm *Text Detection in a 2-D Plot*
Input: Binarized Image matrix Segmented with Regions
Output: Strings set for OCR input.

1. **Initialize:** strings-set $\leftarrow\phi$; cc-set $\leftarrow\phi$; $\delta \leftarrow 20\%$ of the image area
2. For X/Y Region
3. cc-set $\leftarrow$Connected Component Labeling
4. Detect the horizontal position of the text using horizontal profile for X region and Verticle profile for Y region.
5. separate the text strings using Verticle Profile of the text identified in previous step.
6. Approximate the parameters of Eq.2
7. For Curve Region
8. cc-set $\leftarrow$Connected Component Labeling
9. for each component C_i
10. do if Area$(C_i) > \delta$
11. then cc-set $\leftarrow$ cc-set - C_i
12. for each pair of components $C_i \& C_j$
13. do Calculate $P(R_{ij})$ using Eq(2).
14. if $P(R_{ij}) > 0.05$
15. then string-set $\leftarrow C_i, C_j$
16. strings-set $\leftarrow$ string-set
17. return strings-set

Data Points Detection

Scatter and curve-fitted plots contain geometrical shapes that serve as data points in 2-D plots. Our next step after text block detection is to locate these data points. Isolation of these data points from the curves is essential to perform the shape detection in order to have a mapping with the legend. A reasonable heuristic that the curves have similar pixel width to the width of axes can be utilized to filter the lines in the curve region. We extend the basic idea of the k-median filtering algorithm (Seul, O'Gorman, & Sammon 2000) to perform this operation. To summarize the K-median algorithm, it is a first order pixel-noise filtering technique that

Fig.4 Example of sparse regions (sparse boxes)

5.2 Evaluation Metrics

Standard precision, recall, and F measure are used for evaluating the performance of our proposed method.
Let,

- Pg be the set of all pseudocodes,
- Pr be the set of detected pseudocodes,
- Pg ∩Pr are correctly detected pesudocodes,
- F measure the combined precision and recall and is the harmonic mean of precision and recall.

$$precision = \frac{|\mathrm{Pg} \cap \mathrm{Pr}|}{|\mathrm{Pr}|}$$

$$recall = \frac{|\mathrm{Pg} \cap \mathrm{Pr}|}{|\mathrm{Pg}|}$$

$$F1\ measure = 2.\frac{\mathrm{precision.\,reccall}}{\mathrm{precision} + \mathrm{recall}}$$

The proposed methods are exprimented on the test dataset DS2 to measure the perfomance metrics. It has been found that the rule based method produced high precesion with a low recall. The combined approach produced a good performance indicator with overall improvement by 18%.

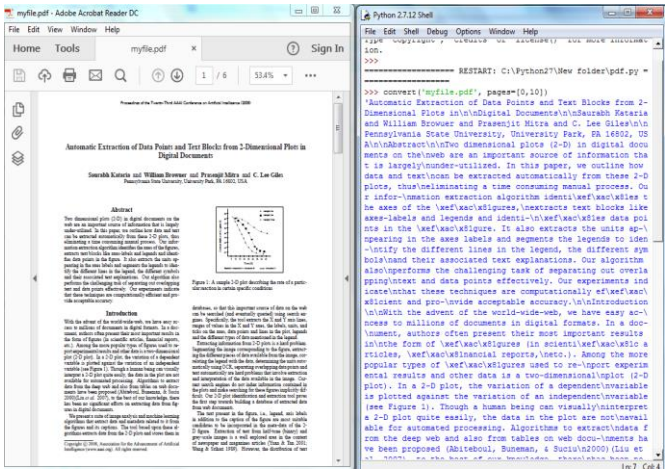


Fig. 5 Screenshots showing PDF to TEXT extraction

6. Conclusion

Algorithms are important and essential part of computational science research work which is regularly published in scholarly articles. In digital libraries, being able to recognize and catalog these algorithms would offer a number of applications like algorithm searching, discovering and analyzing. We have proposed a method to develop an algorithm searching engine. The proposed system analyzes a document to discover any algorithms that may be there in the document. If any algorithm is discovered in the document, the document text is further analyzed to extract additional information about the algorithm. Machine learning and rule based approaches are proposed to discover algorithm representation. Finally we have explained how algorithms found and their associated metadata are indexed and made available for searching through a text query interface. The methods are evaluated and the performance indicators are measured.

References

[1] S Bhatia, Prasenjit Mitra, C. Lee Giles (2016), "AlgorithmSeer: A System for Extracting and Searching for Algorithms in Scholarly Big Data," IEEE transaction on Big Data,vol. 2, no. 1,pp.3-17.

[2] S. Bhatia, S. Tuarob, P. Mitra, and C. L. Giles, "An algorithm search engine for software developers," in Proc. 3rd Int. Workshop Search-Driven Develop. Users, Infrastructure, Tools, Evaluation, pp. 13–16, 2011.

[3] S. Bhatia, P.Mitra, and C. L. Giles, "Finding algorithms in scientific articles," in Proc. 19th Int. Conf.WorldWideWeb, pp. 1061–1062,2010.

[4] H.-H. Chen, L. Gou, X. Zhang, and C. L. Giles, "Collabseer: A search engine for collaboration discovery," in Proc. 11th Annu. Int. ACM/IEEE Joint Conf. Digital libraries, pp. 231–240, 2011.

[5] P. Chiu, F. Chen, and L. Denoue, "Picture detection in document page images," in Proc. 10th ACM Symp. Document Eng.,pp. 211–214, 2010.

[6] M. Khabsa, P. Treeratpituk, and C. L. Giles, "Ackseer: A repository and search engine for automatically extracted acknowledgments from digital libraries," in Proc. 12th ACM/IEEE-CS Joint Conf. Digital Libraries, pp. 185–194,2012.

[7] S. Kataria, W. Browuer, P. Mitra, and C. L. Giles, "Automaticextraction of data points and text blocks from two-dimensional plots in digital documents," in Proc. 23rd Nat. Conf. Artif. Intell. -Volume 2, pp. 1169–1174,2008.

[8] H. Li, I. Councill, W.-C. Lee, and C. L. Giles, "Citeseerx: An architecture and web service design for an academic document search engine," in Proc. 15th Int. Conf. World Wide Web, pp. 883–884, 2006

[9] Y. Liu, K. Bai, P. Mitra, and C. L. Giles, "Tableseer: Automatic table metadata extraction and searching in digital libraries," in Proc. 7th ACM/IEEE-CS Joint Conf. Digital Libraries,, pp. 91–100, 2007.

[10] S. Mandal, S. P. Chowdhury, A. K. Das, and B. Chanda, "Automated detection and segmentation of table of contents page from document images," in Proc. 12th Int. Conf. Image Anal. Process, pp. 213–218, 2003.

[11] S. Tuarob, S. Bhatia, P. Mitra, and C. L. Giles, "Automatic detection of pseudocodes in scholarly documents using machine learning," in Proc. 12th Int. Conf. Document Anal. Recog. pp. 738–742, 2013.

[12] S. Tuarob, P. Mitra, and C. L. Giles, "Improving algorithm search using the algorithm co-citation network," in Proc. 12th ACM/IEEECS Joint Conf. Digital Libraries, pp. 277–280, 2012.

[13] S. Tuarob, P. Mitra, and C. L. Giles, "A hybrid approach to discover semantic hierarchical sections in scholarly documents," inProc.13th Int. Conf. Document Anal. Recog, pp. 1081–1085, 2015.

[14] J. Wang, "Mean-variance analysis: A new document ranking theory in information retrieval," in Proc. 31th Eur. Conf. IR Res. Adv. Inform. Retrieval, pp. 4–16, 2009.