Efficient Exploration of Algorithm in Scholarly Big Data Document

AkshataSanas
JSPM Narhe Technical Campus, Narhe RajarshiShahu School Of Engineering &Research
NTC, Pune.

Abstract:

Computer science and many of its applications are about developing, analysing, and applying algorithms. Efficient solutions to important problems in various disciplines other than computer science usually involve transforming the problems into algorithmic ones on which standard algorithms are applied. Scholarly Digital documents are increasing day by day. To automatically find and extract these algorithms in this vast collection of documents that enable algorithm indexing, searching, discovery, and analysis. AlgorithmSeer, a search engine for algorithms, has been investigated as part of CiteSeerX with the intent of providing a large algorithm database. A novel set of scalable techniques used by AlgorithmSeer to identify and extract algorithm representations in a heterogeneous pool of scholarly documents is proposed. Along with this, anyone with different levels of knowledge can access the platform and highlight portions of textual content which are particularly important and relevant. The highlighted documents can be shared with others in support of lectures and self-learning. But the high-lighted part of text cannot be useful to different levels of learners. We solve the problem of predicting new highlights of partly highlighted e-learning documents.

Keyword: AlgorithmSeer, CiteSeerX, scholarly document.

Introduction

Computer science is about developing, analysing, and applying algorithms. Efficient solutions to important problems in various disciplines other than computer science usually involve transforming the problems into algorithmic ones on which standard algorithms are applied. Furthermore, a thorough knowledge of state of- the-art algorithms is also crucial for developing efficient software systems. Standard algorithms are usually collected and cataloged manually in algorithm textbooks, encyclopaedias, and websites that provide references for computer programmers. While most standard algorithms are already cataloged and made searchable, especially those in online catalogs, newly published algorithms only appear in new articles. The explosion of newly developed algorithms in scientific and technical documents makes it infeasible to manually catalog these newly developed

algorithms. Manually searching for these newly published algorithms is a nontrivial task. Researchers and others who aim to discover efficient and innovative algorithms would have to actively search and monitor relevant new publications in their fields of study to keep abreast of latest algorithmic developments. The problem is worse for algorithm searchers who are inexperienced in document search.

We would like to have a system that automatically discovers and extracts algorithms from scholarly digital documents. Such a system could prove to facilitate algorithm indexing, searching, and a wide range of potential knowledge discovery applications and studies of the algorithm evolution, and presumably increase the productivity of scientists. Since, algorithms represented in documents do not conform to specific styles, and are written in arbitrary formats, this becomes a challenge for effective identification and extraction.

Problem Definition:

Reading documents with huge amount of data and find the algorithm in whole document as well as highlighted part. So, to this problem of predicting new highlights of partly highlighted electronic learning documents. To identify and extract algorithm representations in a heterogeneous pool of scholarly documents.

Objectives

The current study in taken in hand keeping in mind the following objective: -

The main objective is to provide highlighted documents to new users and extract algorithm in pdf. Reading new documents will be time consuming and it will can lead to confusion. Finding new document and going through it is time consuming concept and to save this time we are using documents which are already highlighted. This will help new user to read those important part of documents and will save time and easily find the extract algorithm in pdf document. As users give time to read and understand so we will use that by using already highlighted documents by them and will help new users to save time.

System design:

Data Flow Diagram 0:

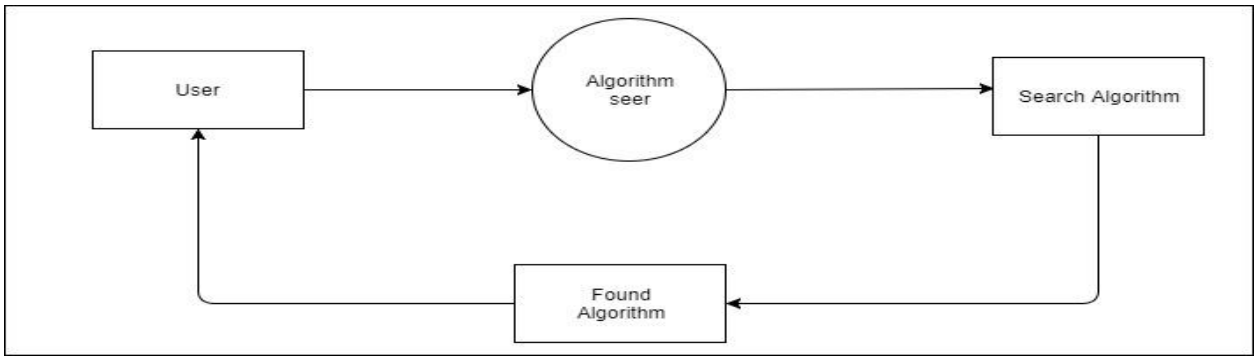


Fig 1: DFD level 0

Data Flow Diagram 1:

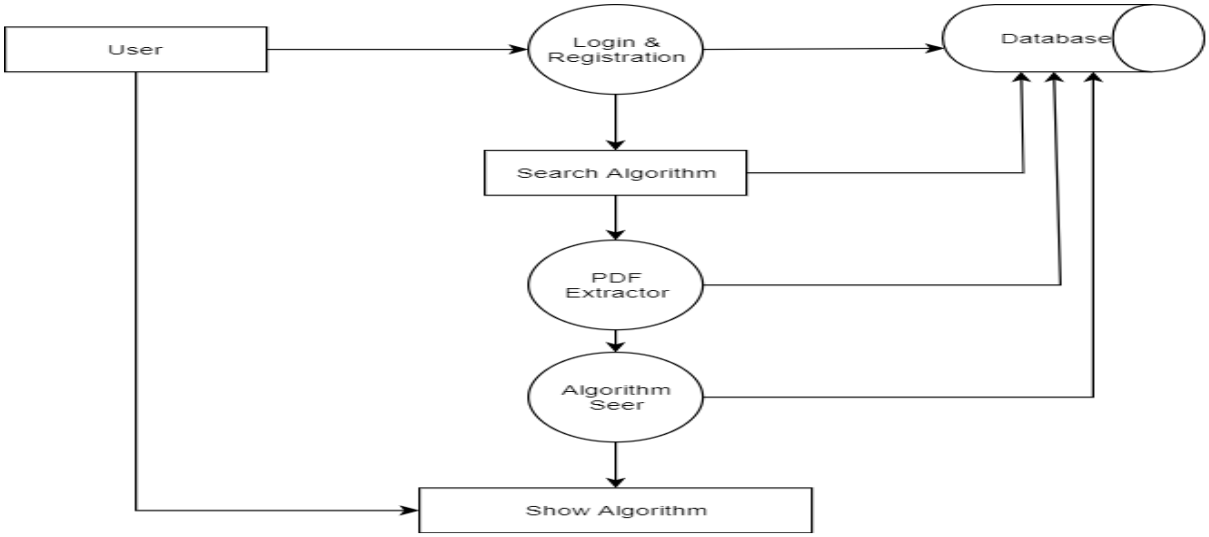


Fig 2 :Dfd level 1

Data Flow diagram 2:

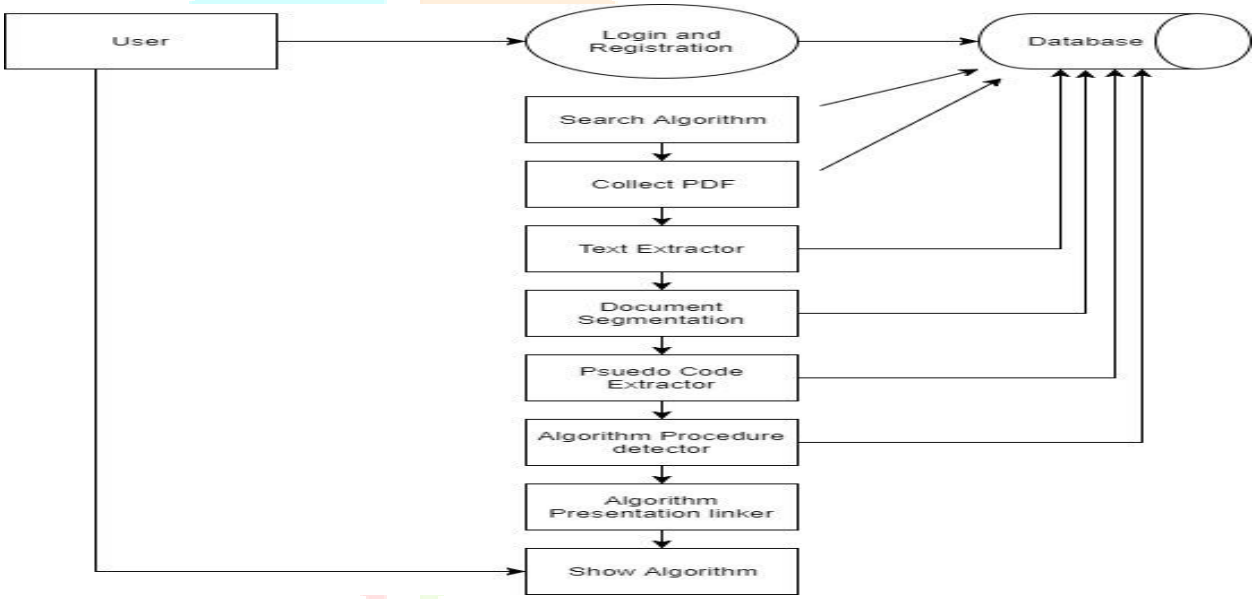
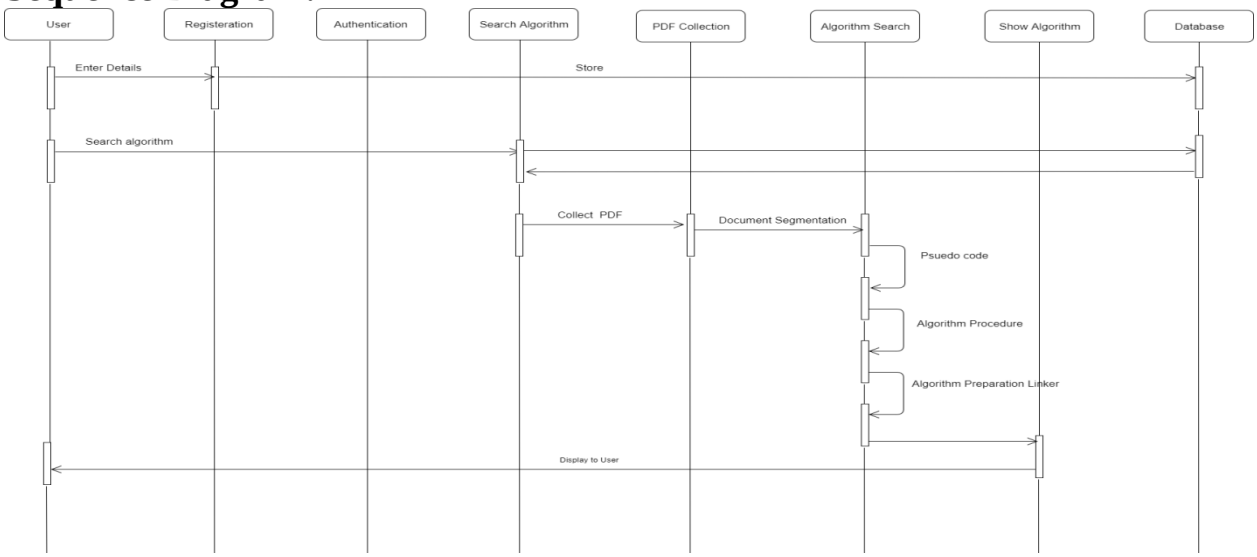
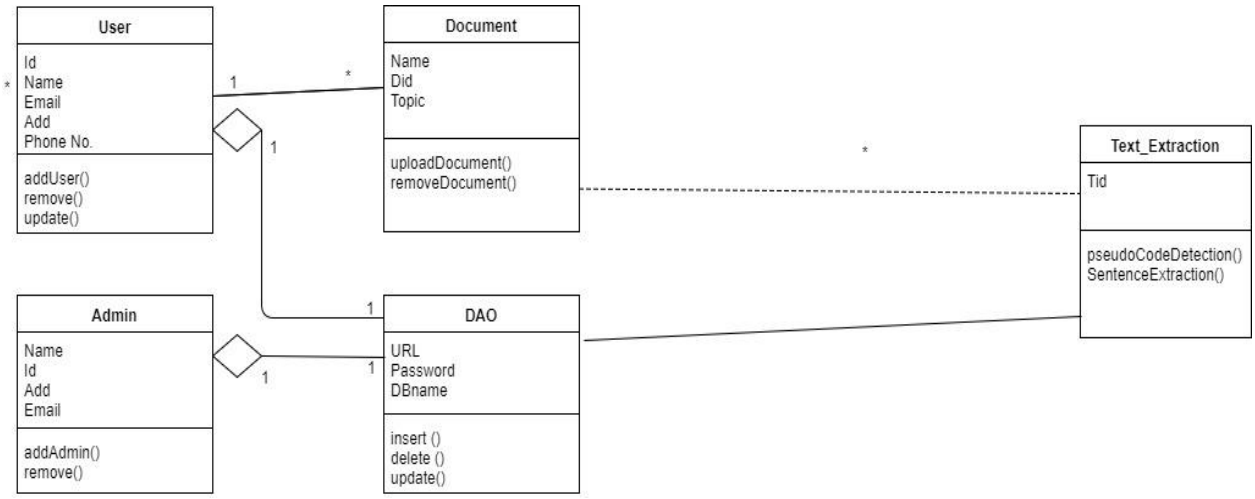


Fig 3 :Dfd level 2

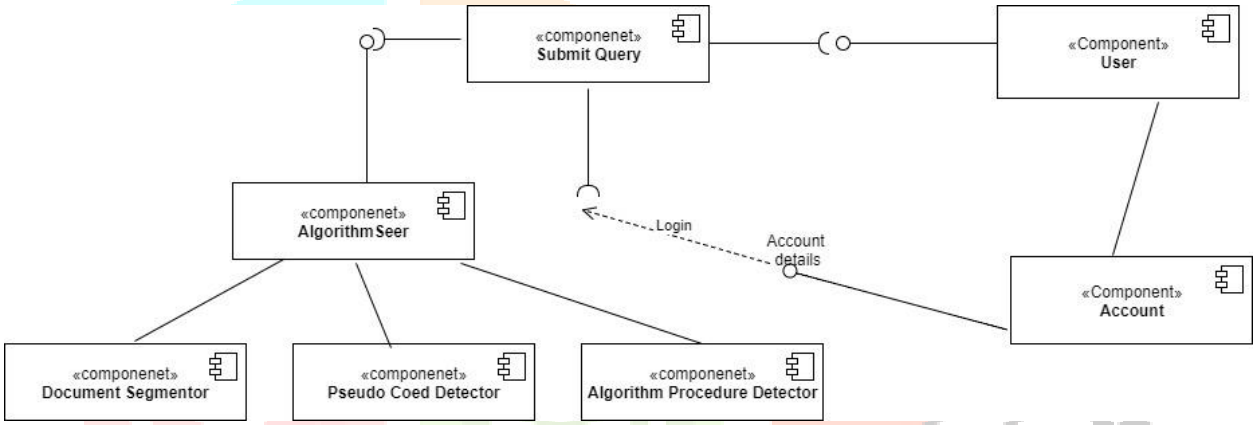
Sequence Diagram:



CLASS DIAGRAM



COMPONENT DIAGRAM



Related Works

HYPOTHESES

1- In this, we are developing a system that helps a user to search algorithm and user highlighted points.

2- To identify algorithm representations in a document.

3- To extract algorithm representations in a document.

4- To facilitate algorithm indexing, searching

5- To increase the productivity of scientists

DELIMITATION OF THE STUDY

1- In this paper we used extract the algorithm from pdf documents.

2- And then we find the list of algorithms.

DESIGN OF THE STUDY

In this paper, when some new concept is build, new algorithm are build. This n numbers of algorithm are stored in some document. Computer science student are search some algorithm, he/she could not find perfect algorithm. To make easy search we are building the AlgorithmSeer system, having the capability to exact search algorithm.

SAMPLE OF THE STUDY

In this, we are developing a system that helps a user to find the algorithms.

TOOLS USED

Software Requirement: A social networking service is an online platform which people use to build social relation with other people who share similar personal or career interests, activities, backgrounds or real-life connections. Social networking services are Internet-based applications.

- Operating System: Windows 95/98/2000/XP
- Front end : HTML, JSP, CSS
- Backend: MySQL
- JDK 1.8

Hardware Requirement: The hardware design of the system includes designing the hardware units and the interface between those units.

- Processor – Pentium –III
- Speed – 2.4 GHz
- RAM - 256 MB (min)
- Hard Disk - 20 GB

ALGORITHM

Steps:

1) Plain text is extracted from the PDF file.

2) Then, three sub-processes operate in parallel, including document segmentation, PC detection, and AP detection.

- 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 then identifies APs.

3) Apply text classification for extracting highlighted text.

4) After PCs and APs are identified, the final step involves linking these algorithm representations referring the same algorithms together.

5) The final output would then be a set of unique algorithms with highlighted document.

System Workflow

The user searching some algorithm on the CiteSeerX dataset, this site gives so many document with his relevant search. On the search result all document index with his best ranking and extract all data related document. This document are in the form of synopses, on the search keywords find the algorithm and gives output to user.

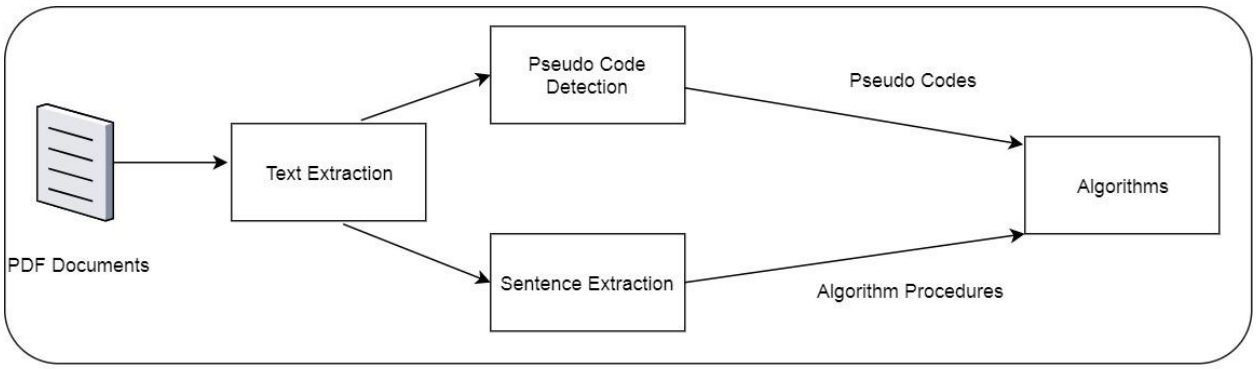


Fig 2: Propose system Architecture Diagram

OUR APPROACH

S={s,e,i,F,o}

- S represents our proposed system.
- s represents start state of the system.
- i represents input of the system i.e. PDF Documents.
- o represents output of the system i.e. set of unique algorithm.
- e represents end state of the system.
- F ={f1,f2,f3,f4,f5}
    - represents Functions of the system.
    - f1=Document_Segmenter.
    - f2=Pseudo_code_detector.
    - f3=Text_cleaner.
    - f4=Sentence_Extracter.
    - f5= algo_procedure_detector.
- The efficiency of the sparse box extraction method is evaluated in two perspectives: coverage and accuracy. Given a set of sparse boxes B extracted from a document d, the coverage is defined as following:

- The accuracy evaluation quantifies how precisely eachPC is cut into a sparse box

FUTURE SCOPE:

1. Our system is useful in Computer Science.
2. Helpful to algorithm searchers.
3. It is useful in forming web-based scientific literature digital library.

CONCLUSION:

To find the algorithm in to many crowed of data related to computer science document and give proper output to user searching keyword.

Reference:

- S. Kataria, W. Browuer, P. Mitra, and C. L. Giles. Automatic extraction of data points and text blocks from 2-dimensional plots in digital documents. In Proceedings of the 23rd national conference on Artificial intelligence - Volume 2, AAAI'08, pages 1169–1174. AAAI Press, 2008.
- J. B. Baker, A. P. Sexton, V. Sorge, and M. Suzuki. Comparing approaches to mathematical document analysis from pdf. ICDAR '11, pages 463–467, 2011.
- S. Bhatia, P. Mitra, and C. L. Giles. Finding algorithms in scientific articles. WWW '10, pages 1061–1062, 2010.
- D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. J. Mach. Learn. Res., 3:993–1022, Mar. 2003.
- J. Kittler, M. Hatef, R. P. W. Duin, and J. Matas. On combining classifiers. IEEE Trans. Pattern Anal. Mach. Intell., 20(3):226–239, Mar. 1998.
- S. Bhatia, S. Tuarob, P. Mitra, and C. L. Giles. An Algorithm Search Engine for Software Developers. 2011.
- TA. Asuncion, M. Welling, P. Smyth, and Y. W. Teh. On smoothing and inference for topic models. In Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence, UAI '09, pages 27–34, Arlington, Virginia, United States, 2009. AUAI Press.
- CP. Chiu, F. Chen, and L. Denoue. Picture detection in document page images. DocEng '10, pages 211–214, 2010.
- S. Bhatia and P. Mitra. Summarizing figures, tables, and algorithms in scientific publications to augment search results. ACM Trans. Inf. Syst., 30(1):3:1–3:24, Mar. 2012.
- G. W. Klau, I. Ljubi, P. Mutzel, U. Pferschy, and R. Weiskircher. The fractional prize-collecting Steiner tree problem on trees. Springer.