MJRET
Open Access
Multidisciplinary Journal of Research in Engineering and Technology
Journal homepage: www.mjret.in
ISSN:2348 - 6953

ALGOMINE: A SYSTEM FOR EXTRACTING AND SEARCHING FOR ALGORITHMS IN SCHOLARLY BIG DATA

NishantBobade, Kiran Kshirsagar, Suraj Inamdar, Priyanka More

B.E. Student, Department of Information Technology,
GSMCOE Balewadi, Pune, India

Abstract:*Algorithms are normally published in scholarly articles, especially in the computational sciences and related fields. The ability to mechanically find and retrieve these algorithms in this increasingly wide collection of scholarly digital articles would enable algorithm discovery, indexing, analysis, and searching. In recent time, AlgoMine, a search engine for algorithms, has been investigated as module of CiteSeerX with the intent of serving a large algorithm database. Now, over 200,000 algorithms have been mined from over 2 million scholarly articles. This paper proposes a new set of scalable techniques used by AlgoMine to search and retrieve algorithm representations in a different pool of scholarly articles. Specifically, hybrid machine learning approaches are proposed to discover algorithm presentation. Then, techniques to extract textual metadata for each algorithm are discussed. Conclusively, a demonstration version of AlgoMine that is built on Solr/Lucene open source indexing and search system is presented.*

Keywords: *Ensemble Machine Learning, Algorithm Search Engine, Scholarly Big Data*

A. INTRODUCTION

Computer science and many of its applications are about developing, analyzing, 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 algorithm seer applied. Furthermore, a thorough knowledge of state-of-the-art algorithms is also crucial for developing efficient software systems.

A significant number of scholarly articles in computer science and related disciplines contain high-quality algorithms developed by researchers. With dozens of new algorithms being reported in these conferences every year, it would be useful to have automated systems that

efficiently identify, extract, index and search this ever increasing collection of algorithmic innovations. Such systems could provide an alternative source for researchers and software developers looking for cutting-edge algorithmic solutions to their problems.

Standard algorithms are usually collected and cataloged manually in algorithm textbooks 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. Ideally, 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.

Identifying and extracting various informative entities from scholarly documents is an active area of research. For algorithm discovery in digital documents, Bhatia, etal, have described a method for automatic detection of *pseudo-codes* (*PCs*) in Computer Science publications. Their method assumes that each PC is accompanied by a caption Such a PC can then be identified using a set of regular expressions to capture the presence of the accompanied caption.

However, such an approach is limited in its coverage due to reliance on the presence of PC captions and wide variations in writing styles followed by different journals and authors. We found that 25.8% (71 out of 275) of the PCs did not have accompanied captions, and would remain undetected by their proposed approach. Furthermore, even though PCs are commonly used in scientific documents to represent algorithms, a majority of algorithms are also represented using *algorithmic procedures* (*APs*). An algorithmic procedure is a set of descriptive algorithmic instructions and differs from a PC in the following ways:

B. LITERATURE SURVEY

A. Summarizing figures, tables, and algorithms in scientific publications to augment search results:

The textual metadata extracted from an algorithm representation is inadequate for effective extract, and propose to use the synopsis creation method. These search engines present a thumbnail view of the articles-elements, some articles metadata like the author title of the paper and title, and in lieu of a article snippet, typically, they represent the caption of the article element. While some authors in some disciplines write carefully tailored captions, generally, the author of a article assumes that the caption will be read in context of the text in the article. When the caption is presented out-of context as in a article-element-search-engine result, it may not contain enough information to help the end-user understand what

the content of the article-element is. Consequently, end-users examining article-element search results would want a short “synopsis ” of this information represented along with the article- element. Having access previleges to the synopsis allows the end-user to quickly understand the content of the article-element without having to read and download the entire article.[1]

B. Finding algorithms in scientific articles:

It described a method for automatic detection of *pseudo-codes.* Their method assumes that each PC is accompanied by a caption. Such a PC can then be identified using a set of regular expressions to capture the presence of the accompanied caption. Algorithms are an integral part of computer science literature. However, none of the current search engines offer specialized algorithm search facility. We represent a vertical search engine that searches the algorithms present in articles and retrives and indexes the related metadata and textual description of the identified algorithms. This algorithm specific information is then utilized for algorithm ranking in response to user queries. Experimental results show the superiority of our system on other popular search engines.[2]

C. A generalized topic modeling approach for automatic document annotation:

Automatically textually enriches metadata records, to extract more meaningful textual information for PCs. Ecological and environmental sciences have become more advanced and complex, requiring experimental and observational data from multiple times, places, and thematic scales to verify their hypotheses. Over time, like this data have not only enlarged in amount, but also in diversity and heterogeneity of the data origins that spread throughout the world. This diversity poses a huge challenge for scientists who have to by hand search for desired data.[3]

3. IMPLEMENTATION DETAILS

A prototype of an algorithm search engine, *Algomine*, is presented the high level of the proposed system. First, scholarly articles 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. Note that, even though the methods presented in this paper are developed to handle algorithms in scholarly digital articles, the methods could be generalized to other information sources that contain textual representations of algorithms such as pseudo-codes found in web-pages (e.g. Stack Overflow, Wikipedia, etc.). Algorithm extraction in html is different from algorithm extraction in free text, though there are some similarities. For example, WebPages are usually composed in HTML which provides markup tags (e.g. <tr>, <td>, etc.) that can be utilized to detect boundaries of article elements, while such functionalities do not exist in free text data. It is also possible that the ideas presented in this paper could be developed for other article elements such as tables and diagrams, as long as textual information from such entities could be extracted.

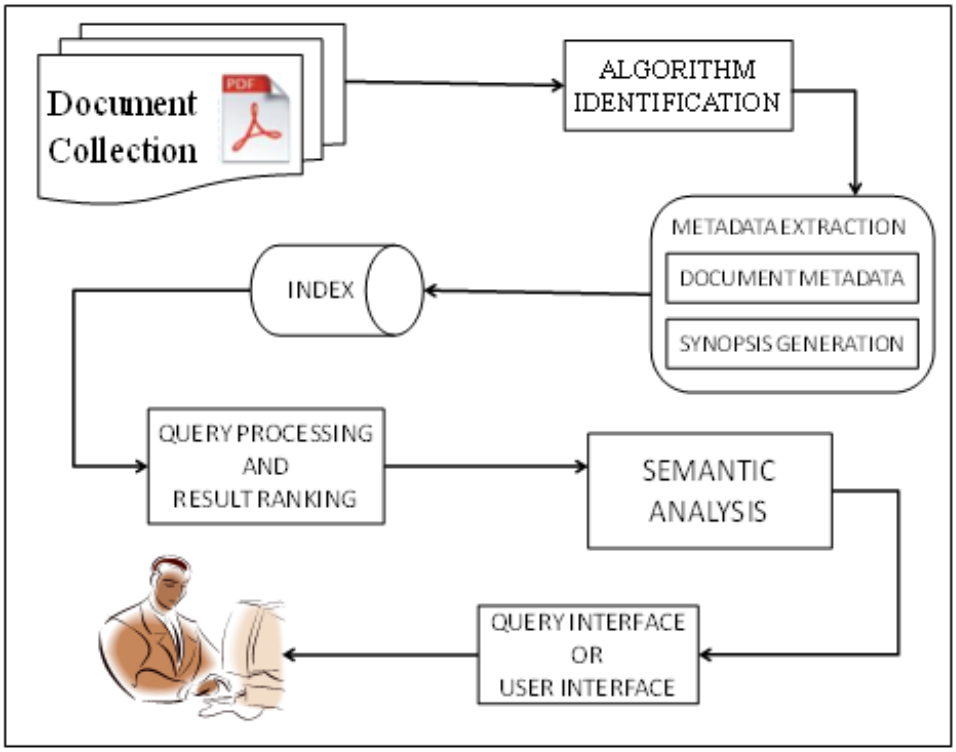

Fig 1. System Architecture

4. MATHEMATICAL MODEL

S = {U, F, O}

Where

U – Set of users

F- Input file.

O – Set of algorithms which is in file.

Consider a set U consisting of various users,

U= {U1, U2, U3….., Un}

5. DESIGN SCREENSHOTS

Fig 2. Search Algorithm

Search Algorithm: This is Search Algorithm page. In this page Search the algorithm.

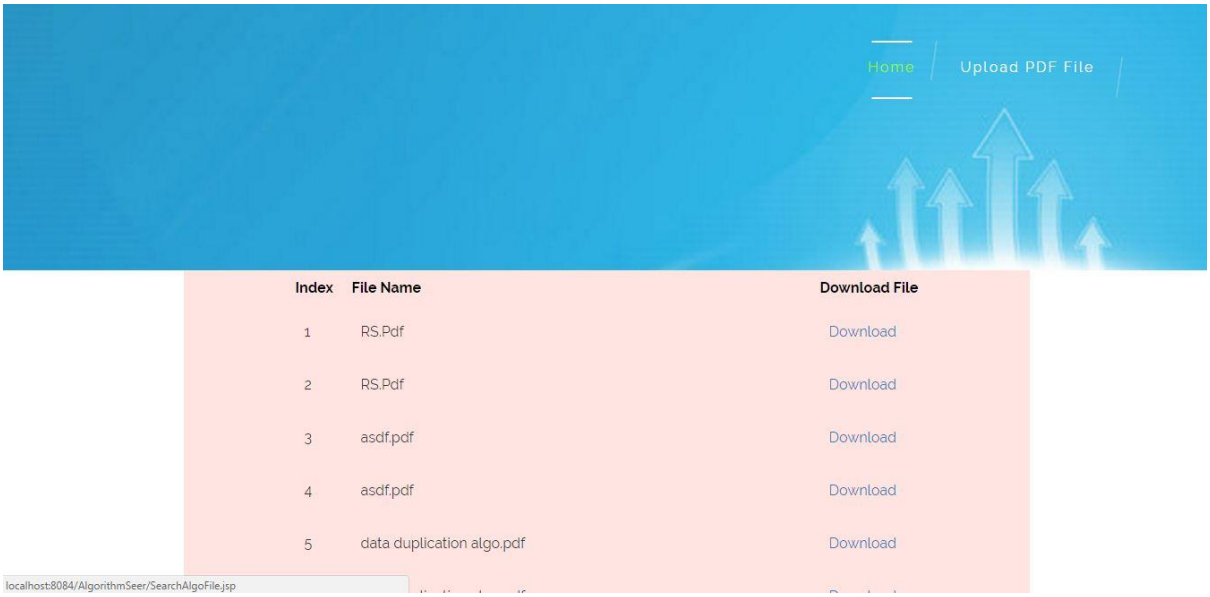

Fig 3. Download Algorithm

Download Algorithm: This is Download Algorithm page. In this page, find the specific algorithm files and this files are downloaded.

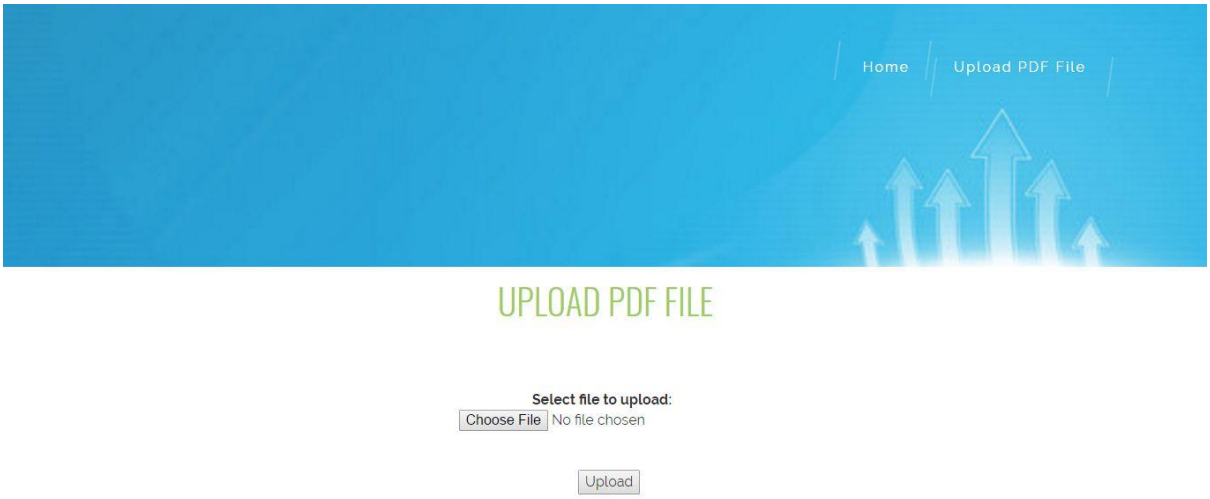

Fig4. Upload PDF File

Upload PDF File: This is Upload PDF File page. In this page uploaded the searchable downloaded file.

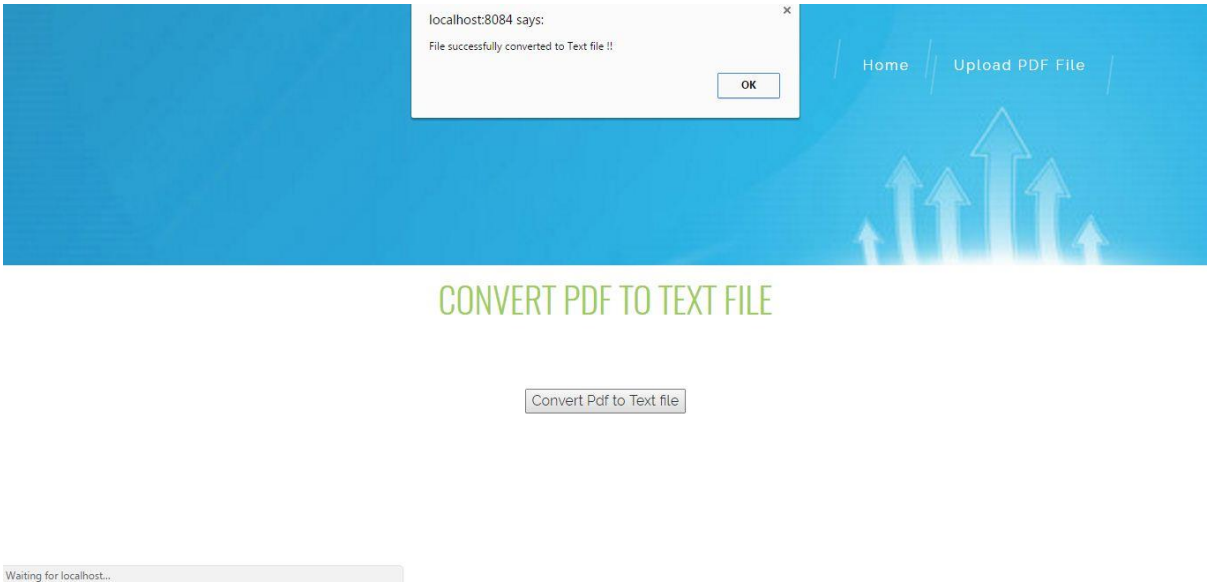

Fig5. Convert PDF to Text File

Convert PDF to Text File: This is Convert PDF to Text File page. In this page Uploaded PDF file is converted into the Text File.

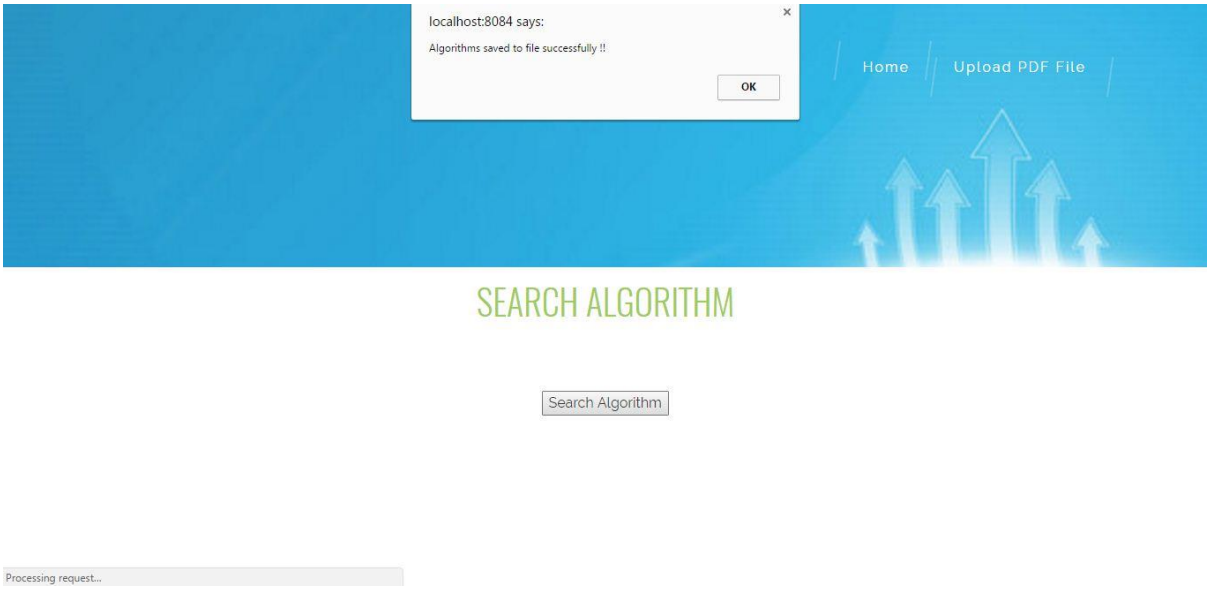

Fig6. Search Algorithm

Search Algorithm: This is Search Algorithm page. In this page Find out the algorithm in our converted text file and this algorithm write in another text file.

6. CONCLUSION

Used using the synopsis generation and document annotation methods to extract textual metadata for pseudo-codes, and finally we explained how algorithms are indexed and made searchable. By Using this project Search the Algorithm related PDF files.

REFERENCES

[1] *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.*

[2] *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.*

[3] *S. Bhatia, P. Mitra, and C. L. Giles. Finding algorithms in scientific articles. WWW '10, pages 1061–1062, 2012 .*

[4] *S. Bhatia, S. Tuarob, P. Mitra, and C. L. Giles. An Algorithm Search Engine for Software Developers. 2011. L. Breiman. Random forest. Machine Learning, 45(1):5–32, 2001.*

[5] *N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. Smote: synthetic minority over-sampling technique. J. Artif. Int. Res., 16(1):321–357, 2002.*

[6] *H.-H. Chen, L. Gou, X. Zhang, and C. L. Giles. Collabseer: a search engine for collaboration discovery. In Proceedings of the 11th annual international ACM/IEEE joint conference on Digital libraries, JCDL '11, pages 231–240, New York, NY, USA, 2011. ACM.*

[7] *P. Chiu, F. Chen, and L. Denoue. Picture detection in document page images. DocEng '10, pages 211–214, 2010.*

[8] *T. Hassan. Object-level document analysis of pdf files. DocEng '09, pages 47–55, 2009.*

[9] *M. A. Hearst, A. Divoli, H. Guturu, A. Ksikes, P. Nakov, M. A. Wooldridge, and J. Ye. BioText Search Engine: beyond abstract search. Bioinformatics, 23(16):2196–2197, 2007.*