International Journal of Research in Advent Technology (IJRAT) Special Issue
E-ISSN: 2321-9637
Available online at www.ijrat.org
National Conference on "Role of Information Technology in Social Innovations"
26th & 27th February 2019

Extract Algorithm, highlighted content and Search Algorithm using Machine Learning and Core NLP with Scholar Big data

Prof. A. D. Khairkar [1], Princy Mishra[2], Ritu Powar[3], Vaishnavi Sonawane[4]
Information Technology [1,2,3,4], Asst. Professor(Bharati Vidyapeeth College of Engineering for Women-Pune)[1], Students[234]
Email: ashkhairkar@gmail.com[1], princymishra1997.5@gmail.com[2], ritupowar79@gmail.com[3], vaishnaviso1997@gmail.com[4]

Abstract-The world of computer science has more concepts with their algorithm. When some new concept is built and then new algorithm should be build. There are n numbers of algorithm stored in some document. Computer science people require searching some algorithm, and it's very difficult to find the relevant algorithm. To overcome this problem we are going to build a system for algorithm searching and extracting highlighted points having best ranked algorithm. We will identify and extract algorithm representations in a heterogeneous pool of intellectual documents. In this system, the main formation/weight of pdf documents is calculated by TF-IDF technique, each and every word having its weight based on that ranked up technique is going to work. For developing purpose the real time pdf data will be downloaded from CiteSeerX site.

Index Terms – AlgoExtract; CiteSeerX; Scholarly document; Stemming; Regular Expression; TF-IDF; Stop word elimination; NLP.

1. THE MAIN TEXT

Computer science is about creating, evaluating, and implementing algorithms. There are efficient number of problems in areas other than Computer Science which require study of algorithms to reach out to the perfect solution to their problem.

Furthermore, an efficient knowledge of algorithms is also important for developing relevant software systems. Usually people can find the algorithms collected in some books, internet sources or encyclopedias. While the ordinary algorithms and pseudo codes are made available online but the recently published pseudo codes are available only in the recent magazines or research papers. The outbreak of new algorithms in research papers or magazines makes it non-functional to manually categorize them[1]. Manually exploring for these recent algorithms is a challenging work. Amateur developers and others who wish to learn about pseudo codes need to search for algorithms over Internet or the magazines/papers which are sometimes out of their reach.

We wish to develop a system that can automatically identify and cite algorithms/pseudo-codes from scientific papers as well as magazines. Such a system will ease algorithm exploring and increment the yield of amateur developers. Since, algorithms represented in documents can be in different patterns or can be represented in different ways, so this task becomes a challenge while identifying the algorithms.

2. RELATED WORKS

As literature on scholarly data extraction is huge, hence only works closely related to ours are discussed.

2.1. *Element Extraction from Scholarly documents*

Analyzing and deriving information such as algorithms/pseudo-codes[1], mathematical expressions [2], tables, figures [9], [3], and index(contents) from documents has been adequately thought-through. S. Tuarob, et al. [1], deliberated machine learning based extraction of algorithm from pdf documents. Kataria, et al. [2], used image processing and Optical Character Recognition (OCR) approaches for automatic

extraction of data points and text blocks from 2-D plots.

2.2. *Search Systems for Scholarly Information*

Apart from renowned web search engines such as Google and Microsoft's Bingo, various vertical search engines have been proposed. CiteSeerX, got developed as a scientific digital library which could immediately index the scholarly documents mainly in the field of Computer Engineering[1]. Liu, et al. proposed TableSeer, a tool which could automatically extract tables from scholarly digital documents [10] using vector-space model ranking algorithm, to rank the search results. The search results can be infected with insignificant details. For example, in CiteSeer X very less searches for the required query "Classification Algorithms" were relevant, while the other results were documents consisting of those searched terms. Furthermore, these search engines would return the whole documents (papers or websites) as search results, which required the users to put extra, futile job in reading the whole documents to find the desired algorithms/pseudo-code.

3. SYSTEM ARCHITECTURE

Figure 1 shows the detailed flow of our System. In this user can upload PDF and algorithm name as an input. Using core NLP technique and Content-based filtering, given PDF file or PDFs in our database will be processed. We are going to perform operations like stemming, stop words removal and parsing technique. After this semantic similarity check will be performed on a word or concept level as well as document or text level. Based on the similarity check TFIDF values will be calculated from words present in an already uploaded document. Based on TFIDF values the required algorithm will be returned to the user in the form of mini-synopsis. This mini-synopsis will contain abstract, desired algorithm and highlighted points from PDF.

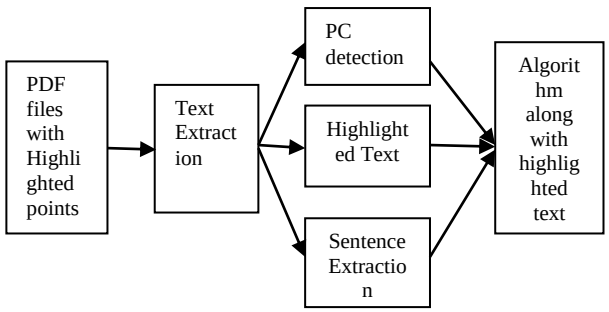

Fig. 1. System Architecture

3.1. *TFIDF Technique:*

To compute the TF-IDF value for a one document, we take that document and calculate the TF-IDF score for each unique word without stop words. For each unique word, or term,

1. Term frequency: this is used for measuring how frequently particular term appears in documents.

$$\mathrm{TF}(\mathrm{Term}, \mathrm{Document}) = \frac{\text{No. of times term appeared in doc}}{\text{Total no. of words}}$$

Eq. (1)

2. Inverse Document Frequency: this is used for measuring how frequently particular term appears in all documents.

$$\mathrm{IDF}(\mathrm{Term}) = \begin{cases} 0 & \text{If term doesn't appear in any doc} \\ \ln\left(\frac{\text{No. of docs}}{\text{No. of docs containing term}}\right) & \text{Otherwise} \end{cases}$$

Eq. (2)

3. On equation 1 and 2 calculate overall TFIDF of word,

$$\mathrm{TF} - \mathrm{IDF} = \mathrm{TF} * \mathrm{IDF}$$

ACKNOWLEDGMENTS

We take this opportunity to thank our project guide Prof. A. D. Khairkar and Head of the Department Prof. Dr. D. A. Godse for their valuable guidance and for providing all the necessary facilities, which were indispensable in the completion of this project report. We are also thankful to all the staff members of the Department of Information Technology of Bharati Vidyapeeth College of Engineering for Women Pune for their valuable time, support, comments, suggestions and persuasion. We would also like to thank the institute for providing the required facilities, Internet access and important books.

REFERENCES

[1] S. Tuarob , Sumit Bhatia, P. Mitra and C. Lee Giles. AlgorithmSeer: A System for Extracting and Searching for Algorithms in Scholarly Big Data. IEEE transactions on big data, 2016.

[2] 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.

[3] 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.

[4] S. Bhatia, P. Mitra, and C. L. Giles. Finding algorithms in scientific articles. WWW '10, pages 1061–1062, 2010.

[5] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. J. Mach. Learn. Res., 3:993–1022, Mar. 2003.

[6] 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.

[7] S. Bhatia, S. Tuarob, P. Mitra, and C. L. Giles. An Algorithm Search Engine for Software Developers. 2011.

[8] 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.

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

[10] Y. Liu, K. Bai, P. Mitra, and C. L. Giles. Tableseer: automatic table

[11] metadata extraction and searching in digital libraries. In Proceedings of the 7th ACM/IEEE-CS joint conference on Digital libraries, JCDL '07, pages 91–100, New York, NY, USA, 2007. ACM.

[12] 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.

[13] G. W. Klau, I. Ljubi, P. Mutzel, U. Pferschy, and R. Weiskircher. The fractional prize-collecting Steiner tree problem on trees. Springer.

[14] S. Mandal, S. P. Chowdhury, A. K. Das, and B. Chanda. Automated detection and segmentation of table of contents page from document images. ICDAR '03, pages 398–, 2003.

[15] S. Tuarob, P. Mitra, and C. L. Giles. Improving algorithm search using the algorithm co-citation network. In Proceedings of the 12th ACM/IEEE-CS joint conference on Digital Libraries, JCDL '12, pages 277–280, New York, NY, USA, 2012. ACM.