International Journal of Research in Engineering and Science (IJRES)
ISSN (Online): 2320-9364, ISSN (Print): 2320-9356
www.ijres.org Volume 12 Issue 9 ǁ September 2024 ǁ PP. 14-17

Optimized PDF Keyword Search with Boyer-Moore and TF-IDF

Joannah Cantoria Argote
Southern Luzon Technological College Foundation Pilar Inc, Philippines

Abstract
A system that can be used to search any keyword in all existing PDF files in a storage device is developed. The system applied the Boyer-Moore Algorithm for efficient keyword matching. The results are ranked by importance, determined Term Frequency – Inverse Document Frequency weight.

The study's findings demonstrate how well the system works to efficiently aid the user in searching through numerous PDF files in different storage locations simultaneously. The algorithm has the ability to quickly find keyword matches and rank the PDF files by perceived keyword importance. Moreover, the system may be integrated in other PDF filet-related projects, such as library management systems or localized offline search engines. The study limits itself to search only for text data in PDF files; it skips non-text data. The study cannot search through password-protected PDF files and encrypted files.

Keywords: *keyword, pdf, searching algorithm, ranking algorithm, boyer-moore algorithm, term frequency – inverse document frequency*

Date of Submission: 24-08-2024 Date of acceptance: 04-09-2024

I. INTRODUCTION

In the digital age, the ability to efficiently search for specific information across multiple documents has become increasingly vital. As the volume of data continues to grow exponentially, particularly in the form of PDF documents, there is a heightened need for advanced search algorithms that can swiftly and accurately locate relevant information. On a global scale, industries, academic institutions, and businesses rely heavily on vast repositories of PDF documents, making the task of searching for specific content within these files both crucial and challenging. Advanced search techniques, like the Boyer-Moore algorithm coupled with Term Frequency – Inverse Document Frequency (TF-IDF) weighting, have emerged as powerful tools to enhance the efficiency and accuracy of keyword searches in such contexts [1].

Internationally, organizations have been leveraging sophisticated search algorithms to manage and extract valuable information from extensive digital libraries. In countries like the United States and Japan, there has been a notable shift towards integrating these algorithms into larger systems, such as digital libraries, online databases, and research repositories [2]. This trend highlights the growing recognition of the importance of effective information retrieval systems in a world that is becoming increasingly data-driven. The adoption of such technologies has not only streamlined the process of data retrieval but has also set a standard for how digital information is managed and accessed [3].

In the Philippines, the demand for efficient information retrieval systems is also on the rise, particularly within academic and governmental institutions. With a significant amount of data stored in PDF format, there is a pressing need for systems that can quickly and accurately search through these documents [4]. However, many existing systems lack the sophistication to rank search results based on the relevance of the keyword to the entire document. This limitation often results in inefficiencies, as users are forced to manually sift through large amounts of irrelevant data. The integration of advanced algorithms like Boyer-Moore and TF-IDF in search systems could greatly enhance the efficiency and effectiveness of information retrieval in the country [5].

In Sorsogon, a province that is steadily advancing in terms of digital literacy and information technology, there is a growing recognition of the need for efficient data management systems. Local government offices, educational institutions, and businesses are increasingly reliant on digital documents for their daily operations. However, the absence of a robust system for searching and ranking keywords within these documents poses a significant challenge. The development of a system that employs the Boyer-Moore algorithm and TF-IDF weighting could offer a solution to this problem, enabling users to quickly find and prioritize relevant information across multiple PDF files stored in various locations.

This project is centered on developing a system designed to search for keywords across all PDF documents stored on a device, using the Boyer-Moore algorithm for efficient keyword matching. The system further enhances the search results by ranking them according to their importance, as determined by the TF-IDF weight. The primary objective is to create a tool that allows users to efficiently navigate large collections of PDF files, aiding in faster and more accurate information retrieval. By focusing on text data within non-password-protected PDF documents, this system addresses the current limitations in existing search tools, offering a practical solution for both local and broader applications.

II. METHODOLOGY

The Rapid Application Development (RAD) methodology was employed in the development of this system to ensure that the software was built quickly and effectively, with constant feedback and iterative improvements. RAD is characterized by its focus on speed, flexibility, and user involvement, making it well-suited for projects where requirements may evolve during the development process [6].

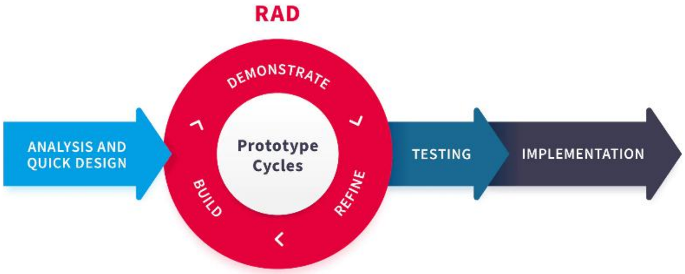

Figure 1. Rapid Application Development

Using the Boyer-Moore Algorithm, the researcher applied the ‘Bad Character Rule’. This rule shifts the search pattern to align the mismatched character in the text with its last occurrence in the pattern, or skips it entirely if absent. This significantly reduces comparisons, enhancing search efficiency [7].

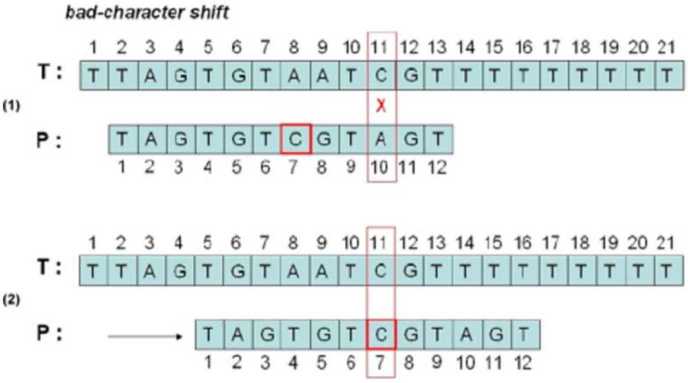

Figure 2. Bad Character Rule – showing how the shift is done

In this system, the ‘Bad Character Rule’ optimized keyword searched within large PDF documents by minimizing unnecessary comparisons. This allowed the algorithm to quickly locate potential matches, making the search process fast and efficient, even with extensive texts.

The Term Frequency-Inverse Document Frequency (TF-IDF) weight is a statistical measure used to evaluate the importance of a keyword within a specific document relative to a collection of documents [8]. In this system, TF-IDF is crucial for ranking files that match after applying the Boyer-Moore Algorithm.

Term Frequency (TF): Measures how often the keyword appears in a document, calculated by dividing the keyword’s occurrences by the total number of words in the document. A higher TF value indicates greater relevance.

$$TF(t) = \frac{number\ of\ times\ t\ appears\ in\ PDF}{number\ of\ matches\ in\ PDF}$$

Inverse Document Frequency (IDF): Assesses how common or rare the keyword is across all documents. It is calculated by taking the logarithm of the total number of documents divided by the number of documents containing the keyword. A higher IDF value signifies that the keyword is rare and thus more significant.

$$IDF(t) = \log e \frac{number\ of\ PDFs + 1}{number\ PDFs\ with\ match\ (t)}$$

TF-IDF Weight Calculation: The TF and IDF values are multiplied to obtain the TF-IDF weight for the keyword in each document, indicating the keyword's importance relative to the entire collection.

$$TF - IDF = TF \times IDF$$

Ranking the Files: Documents with higher TF-IDF weights are ranked higher in search results, helping users quickly identify the most relevant documents based on keyword frequency and significance.

System Analysis

This diagram shows the graphical flow of data using Efficient Keyword Search in PDF Documents Using the Boyer-Moore Algorithm and TF-IDF Weight.

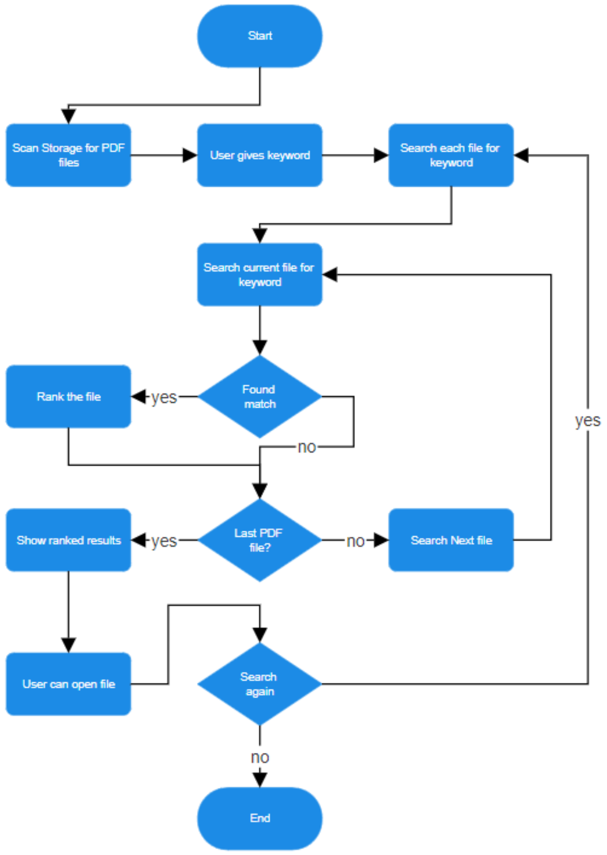

Figure 3. Flow Chart

III. CONCLUSION

Creating a keyword search system using the Boyer-Moore Algorithm and TF-IDF weighting has been a game-changer for efficiently searching and ranking PDF files on a device. The Boyer-Moore Algorithm, especially with its 'Bad Character Rule,' helps the system skip unnecessary comparisons, making the keyword matching process much faster.

By adding TF-IDF weighting, the system doesn't just find keywords; it ranks documents based on how often the keyword appears in a document and how rare it is across all documents. This means users can quickly find the most relevant documents, making it easier to sift through large amounts of data.

This system is particularly useful for handling non-password-protected, non-encrypted text data in PDF documents. It shows great promise for various applications, such as library management systems and localized offline search engines, making it a valuable tool for anyone needing to manage large collections of digital documents.

IV. RECOMMENDATION

To extend the functionality of the system, future enhancements could focus on supporting searches within password-protected and encrypted PDF files. By implementing features that decrypt these files for keyword searches, the system would become more comprehensive and versatile. Additionally, integrating Optical Character Recognition (OCR) technology would allow the system to search within non-text data, such as scanned images or documents stored as images within PDFs. This would broaden the system's capabilities and increase its utility in more varied contexts.

As the system is deployed in environments with larger datasets, optimizing the algorithm to handle even larger volumes of data more efficiently could be beneficial. This might involve refining the indexing process or exploring additional algorithms that could complement the existing Boyer-Moore and TF-IDF approach. Improving the user interface to provide more detailed feedback on search results, such as snippets of text surrounding the keyword match or a more interactive ranking system, could enhance user experience and make the system more intuitive and user-friendly.

Exploring opportunities for integrating this system with other document management or search platforms, especially in institutional or corporate environments, could also be valuable. This could include APIs or plug-ins that allow the system to be used as part of a broader information retrieval ecosystem. Finally, implementing tools for monitoring the system's performance in real-world use and gathering user feedback could provide valuable insights for ongoing improvements, ensuring that the system continues to meet user needs effectively.

REFERENCES

[1]. Pandu Nayak and Prabhakar Raghavan, Stanford University. (n.d.). "Introduction to Information Retrieval". Retrieved from https://web.stanford.edu/class/cs276/19handouts/lecture6-tfidf-1per.pdf

[2]. Glowacka, D. (n.d.). Information Retrieval Lecture 6: Ranked Retrieval. Retrieved from https://glowacka.org/lectures/ir/DATA20021-Lecture6-web.pdf

[3]. MDP. (n.d.). Information Retrieval In Text-Based Document Using Boyer Moore Algorithm. Retrieved from https://jurnal.mdp.ac.id/index.php/jatisi/article/download/2053/799

[4]. Fathima, S. (2024, April 16). TF-IDF Explained: Unlock Keyword Analysis for Your Text Data. MarkovML. Retrieved from https://www.markovml.com/blog/tf-idf

[5]. Rahul. (2024, May 4). TF-IDF - Understanding Term Frequency-Inverse Document Frequency in NLP. Zilliz. Retrieved from https://zilliz.com/learn/tf-idf-understanding-term-frequency-inverse-document-frequency-in-nlp

[6]. Bondar, J. (2024, April 16). The ultimate guide to rapid application development. NIX United. Retrieved from https://nix-united.com/blog/the-ultimate-guide-to-rapid-application-development/

[7]. Boyer, R. S., & Moore, J. S. (1977). A fast string searching algorithm. Communications of the ACM, 20(10), 762-772. : Gusfield, D. (1997). Algorithms on Strings, Trees, and Sequences: Computer Science and Computational Biology. Cambridge University Press.

[8]. Manning, C. D., Raghavan, P., & Schütze, H. (2008). Introduction to Information Retrieval. Cambridge University Press. : Ramos, J. (2003). Using TF-IDF to Determine Word Relevance in Document Queries. Proceedings of the First International Conference on Machine Learning