INTERNATIONAL JOURNAL OF ADVANCE RESEARCH, IDEAS AND INNOVATIONS IN TECHNOLOGY

ISSN: 2454-132X
Impact Factor: 6.078
(Volume 11, Issue 3 - V11I3-1240)
Available online at: https://www.ijariit.com

Deep Search: An Intelligent File Searching through Content Analysis

Nachiket Parjane
nachiketparjane054@gmail.com
G.H. Raisoni College of Engineering and Management, Pune

Kartik Patare
kartikpatare45@gmail.com
G.H. Raisoni College of Engineering and Management, Pune

Rohan Ingle
irohan428@gmail.com
G.H. Raisoni College of Engineering and Management, Pune

Renuka Wakhare
renukawakhareofficial@gmail.com
G.H. Raisoni College of Engineering and Management, Pune

ABSTRACT

Real-time full-text search holds essential value in current digital libraries because it helps users find documents with content rather than names [1]. Users can perform content-based searches that reveal files through the extraction of textual contents within documents and optimize the retrieval process for research databases as well as legal document search and enterprise knowledge management solutions [2]. A full-text search technique-powered document retrieval system, which seeks to create a content-based file searching mechanism, is analyzed within this report. These search systems implement ranking as their main step to evaluate document relevance through the combination of term frequency and document length analysis with inverse document frequency factors [3]. The foundation for improving search accuracy and efficiency depends heavily on knowledge about directory creation as well as score calculation methods. The research also explores performance comparison between Whoosh and Elasticsearch regarding their scaling capabilities and their abilities to index data and respond to search queries and rank results [4]. Whoosh functions best for compact document sets, yet Elasticsearch delivers real-time search functionality for extensive data collections. The final report will present the most effective solution for creating a content-based search system with high performance levels for various application domains.

Keywords— ***Whoosh, Query, Indexing.***

INTRODUCTION

Researchers face major challenges regarding effective retrieval of important information from extensive document repositories in the discipline of information retrieval. File search techniques that use folder paths or metadata and filenames for locating files remain ineffective when users conduct searches based on document content. A content based search system indexes entire document texts as part of its framework which enables users to discover suitable files through keyword or complete text or semantic meaning queries. Development of an efficient content-based search system demands full-text search algorithms which establish document processing for indexing and relevance assessment based on input queries. The search technologies Whoosh and Elastic search function as popular options for this application. The search library Whoosh operates with Python 1 while Elastic search functions as a distributed engine that provides real-time operations for large data scales.

LITERATURE REVIEW

1. Content-Based File Retrieval Systems

The conventional file search procedures depend largely upon data metadata including file titles along with their types and changing dates. The tremendous growth of digital data creates difficulties for users to find files through their individual names alone. Users now find content-based file retrieval systems valuable because these systems enable document searching by reading the actual data contained within files.

Key Contributions:

Z. Chen et al. developed a local file system full-text search engine based on Apache Lucene to index Txt Docs and Pdf text documents. Precise keyword search through the system operated with a customizable analyzer and tokenizer feature.

M. Gupta and team developed a desktop search tool with Tf-Idf scoring for relevance ranking because it retrieved effective results when users entered imperfect or incomplete queries.

Rephrase the following sentence to make it direct and easy to understand while also normalizing verbalization when necessary. Several popular open-source tools like Recoll and DocFetcher perform content-based searches effectively although they do not feature real-time searching capabilities or large-scale network scalability.

2. Semantic Search and Natural Language Processing (NLP) in File Retrieval

The main limitation of keyword-based search systems arises from their failure to match conceptually correct queries that do not exactly match stored keywords. Recent research introduces semantic search together with natural language understanding to file retrieval systems in order to understand users' intentions instead of using plain keyword matching.

Key Contributors:

Search systems based on keywords allow users to look up content but struggle with queries which have correct concepts even though they differ from the exact terms. New research introduces semantic search and natural language understanding into file retrieval to understand user intent because keyword matching methods by themselves are inadequate.

J. Singh together with his team developed an indexing method which merges keyword approaches with natural language processing infrastructure to achieve better recall and precision results in datasets with overlapping document terminologies.

Local desktop applications demonstrate restricted usability of Elastic Search with NLP plugins which implement scalable semantic search solutions for structured and unstructured data because these solutions require high resources and complexity management. The development of adaptive information systems for industrial and community usage needs further exploration.

3. File Indexing and Format Compatibility

Any content-based search system requires an accurate and efficient motor that indexes different file types. Indexing technology generates the ability to access content speedily and efficiently but its operational strength directly affects both program performance and user experience. The system needs to manage numerous file types including PDFs and Word files together with Excel files as well as PPTs and plain text combined with code files while ensuring precise content retrieval by search functions.

Key Contributors

The researchers at L. Kumar introduced an indexing system that utilizes Apache Tika to extract content from multiple document types. According to the research proper parsing occurred across more than 30 distinct file formats while emphasizing the need to include solid format parsers for supporting different user requirements.

The research of S. Patel and co-authors focused on evaluating indexing performance together with storage optimization for environments containing large document volumes through applications of inverted indexing and delta compression methods.

Numeric, text, email, and document indexing are basic functions found in Windows Search and mac OS Spotlight although these systems fall short when it comes to specialized developer capabilities and comprehensive content analysis

REFERENCES

[1] D. Cutting and J. Pedersen, "Optimizations for dynamic inverted index maintenance," in *Proceedings of the 13th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Brussels, Belgium, 1990, pp. 405–411.

[2] E. Voorhees and D. Harman, Eds., *TREC: Experiment and Evaluation in Information Retrieval*. Cambridge, MA: MIT Press, 2005.

[3] J. Zobel and A. Moffat, "Inverted files for text search engines," *ACM Computing Surveys*, vol. 38, no. 2, pp. 6–1–6–56, Jul. 2006.

[4] G. Salton and M. McGill, *Introduction to Modern Information Retrieval*, 1st ed. New York: McGraw-Hill, 1983.

[5] H. Bast, B. Buchhold, and E. Haussmann, "Semantic full-text search with broccoli," in *Proceedings of the 37th International ACM SIGIR Conference on Research and Development in Information Retrieval*, Gold Coast, Australia, 2014, pp. 1265–1266.

[6] J. Dean and S. Ghemawat, "MapReduce: Simplified data processing on large clusters," in *Proceedings of the 6th Symposium on Operating Systems Design and Implementation (OSDI)*, San Francisco, CA, USA, 2004, pp. 137–150.

[7] A. Berger, R. Caruana, D. Cohn, D. Freitag, and V. Mittal, "Bridging the lexical chasm: Statistical approaches to answer-finding," in *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Athens, Greece, 2000, pp. 192–199.

[8] M. Gupta and A. Varma, "Content-based desktop search using term frequency-inverse document frequency (TF-IDF)," *International Journal of Computer Applications*, vol. 92, no. 2, pp. 1–6, Apr. 2014.

[9] A. Roy, K. Rathi, and A. Sharma, "Context-aware semantic search using deep learning," *International Journal of Computer Sciences and Engineering*, vol. 7, no. 2, pp. 112–118, Feb. 2019.

[10] T. White, *Hadoop: The Definitive Guide*, 4th ed. Sebastopol, CA: O'Reilly Media, 2015.