

¹ Mr. Chattar Singh² Dr. Vijay Gupta

Optimizing Information Retrieval in Lucene Using Lexical Chain Analysis and WordNet Integration



Abstract: - Extraction of relevant information from unstructured data is a major challenge in the big data era. This paper proposes a method to extend the search capabilities of Lucene by integrating lexical chain analysis with WordNet. The focus is mainly on enhancing the precision and relevance of results in search. WordNet is an extremely large lexical database that contains all the rich semantic relationships among words, which are used to create lexical chains—sequences of semantically related words, offering insight into the text conceptual structure. Doing the lexical chain analysis increases the complexity of semantics in search queries and with their corresponding semantic relationships to documents, giving real context. It supports diversified search modalities, which contain document attributes such as name, type, size, date, and author, along with content-based searches. A major emphasis of the indexing mechanism would be on keyword frequency and the presence of semantically relevant keywords using lexical chains. In this paper, a methodology and its corresponding algorithms are developed for this integration. This includes explanations of the indexing mechanism and the different search algorithms. By combining the power of Lucene in search, with the semantic depth afforded by WordNet-driven lexical chain analysis, it is the aspiration of this research to significantly enhance information retrieval from unstructured data, addressing evolving requirements across a number of domains and facilitating intuitive access to knowledge repositories.

Keywords: *Lucene, lexical chain analysis, WordNet, information retrieval, unstructured data, semantic search, document indexing, image-based search.*

I. INTRODUCTION

The landscape in the current digital world has experienced remarkable growth in unstructured data, which is challenging. In comparison with structured data, which is highly organized into predefined models like databases and spreadsheets, unstructured data includes a broad array of formats, such as text documents, emails, social media posts, images, and videos, among others, that make up multimedia content. There is no consistent format or schema for this type of data; hence, it is difficult to manage, search, and analyze. This has created exponential growth of such data, thereby overwhelming traditional information retrieval systems with the volume and complexity of the unstructured information to be dealt with [1].

Most traditional search methodologies, being based on keyword matching, have the documents get indexed by using the frequency and occurrence of some set of keywords [2]. This may not be the case with effective methods to work on unstructured data. Related keywords in the search can be ignored and the search may not hit deeper semantic relationships and contextual relevance of the text, thus not bringing the right or incomplete search result. For example, keyword-based search systems tend to be poor at supporting synonyms and polysemy, often resulting in too much irrelevant data returned or missing relevant information [2]. Apache Lucene is a high-performance, full-featured text search engine library implemented in Java [3].

Indexing and searching in Lucene are very powerful, thus making it an ideal tool to embed sophisticated search functionality into the applications. Lucene makes an inverted index, a data structure mapping terms to their locations within a set of documents that enable very fast and efficient retrieval based on keyword queries. Although robust, Lucene's traditional keyword-based approach exhibits quite a few limitations while applied to the complexities of unstructured data [3].

While Lucene is good for structured text data management with searching, it has significant challenges with unstructured data [4]:

¹ Research Scholar, Department of Computer Science, International School of Informatics & Management (ISIM), Jaipur, RTU, Kota, India, chattarsinghr@gmail.com

² Associate Professor & Research Supervisor, Department of Computer Science, International School of Informatics & Management (ISIM), Jaipur, RTU, Kota, India, vijaygupta1@gmail.com

- **Semantic Understanding:** Essentially, Lucene is a keyword matching search mechanism—it has no support for understanding the inherent semantic relationships between words. Such a limitation will fail to process synonyms, antonyms, or contextual meanings. This might not detect documents, even if they were to use other terms to describe the same concept [4].
- **Contextual Irrelevance:** Most of the times, keyword-based searches return documents containing the searched keywords but do not relate to the intention of the user. This leads to irrelevant search results; that is, the appearance of a keyword does not guarantee that the document may be relevant to the posed query [4].
- **Handle Polysemy and Homonymy:** Lucene's keyword-based approach will be constrained by words with multiple meanings and homonymy, letting homonymic words mean the same or sound the same but be different in Ky words. The results often turn up either confusing or imprecise because the context in which the word is used is not understood [4].

These then are the limitations of traditional search that call, therefore, for more advanced one, which would be able to understand and interpret the semantic content of unstructured data. In this context, lexical chain analysis has emerged as a useful method to address these challenges. Words in the lexical chain are semantically related with other words throughout the text. They offer an overall coherent perception of the text by underlining what relates words and concepts to each other. To attempt better relevance and accuracy of search results, lexical chain analysis is sought to be implemented along with the current search engineering [5].

Objective of the Study

This research will, therefore, try to enhance the search capabilities of Lucene through lexical chain analysis so as to take advantage of the richness of semantics deposited within WordNet. WordNet is a huge lexical database that organizes English words in sets of synonyms, called synsets, and establishes several sorts of semantic relations, such as hyponyms, hypernyms, and meronyms. Besides, these lexical chains can be developed with the help of these types of relations. The specific objectives of this study are listed as follows [5]:

- **Elevated Search Precision and Relevancy:** The approach utilizes lexical chain analysis to automatically detect complex semantic linkage within search queries and documents with the ultimate goal of delivering more contextually relevant results than that experienced with more traditional keyword-based approaches [5].
- **Support of Diverse Search Modalities:** Design a support for multiple search modalities including search based on document attributes such as name, type, size, date, and author as well as content-based searches. With such a provision, it validates the performance of users towards more targeted and comprehensive searches [6].
- **Designing an Advanced Indexing Mechanism:** Designing an indexing mechanism that takes into account not only keyword frequency but also emphasizes the presence of semantically sensitive key words, i.e., lexical chains, pointed out above. The emphasis on frequency and semantic relevance shall be two fillip points to the overall effectiveness of the search system [7].

Contributions of the Research

The major contribution of this paper is the innovative combination of lexical chain analysis using the search capabilities provided in Lucene and the semantic knowledge through WordNet[7]. In this context, the work covers methods and algorithms for building lexical chains and ways to index documents and further combines the strong points of both lexical chain analysis and the convenient search capabilities of Lucene to serve information retrieval, particularly with respect to unstructured data.

II. LITERATURE SURVEY

Kanev, A. I., & Terekhov, V. I. (2020, December) [11], talking about semantic search, contrast its precision and recall with respect to Lucene-based search, and consider a new ranking methodology for search engines. Kasmani, F., et al. (2020, March) [14], develop this content-oriented learning to a Content-Based Search Engine for E-Books, aiding exploration with phrase-based indexing. Jin, D., et al. (2020, June) [10] design a lightweight retrieval method based on JDBC and Lucene for rapid content extraction from a database. Lin, J., et al. (2020, July) [11] argue that the open-source search engines can interoperate and propose comparison in a fair manner by the proposed Common Index File Format (CIFF). Wang, D. [12] proposed an advanced-technology-based digital

archive management system with a client/browser/server architecture that is modeled to be cross-platform compatible. Hema, R. (2021) [13] developed a search application for chemical documents with a focus on specialized indexing. Made focusing on revising the searching through text—Chan, H. P. et al. (2021) [14] propose a cloud-based search engine for rich text content.

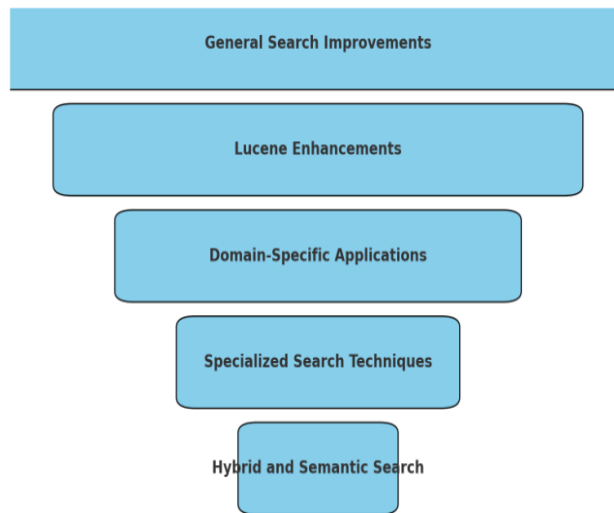


FIGURE 1. Funnel Approach for Literature Review

Safaei, A. (2021) [15], proposes enhancement of the medical image retrieval with a text-based multi-dimensional indexing technique using Lucene. Zouaoui, S., & Rezeg, K. (2021) [16] build a semantic search engine for the Quran based on ontology that simply attains high precision and recall. Haddela, P., et al. (2021) [17] use a Genetic Algorithm to enhance the performance of the queries in Lucene with document classification in Sinhalese datasets, achieving promising accuracy. Azizan et al., 2022 [18] have developed a Lucene-based search engine for Malay Quran translations that has moderate precision and recall. Singh & Gupta, 2022 [19], propose the use of Lucene in web retrieval and plan to extend it into the domain of image search, along with incorporating lexical chains using WordNet. Herein, Liu et al., 2022, October) [20] introduce MMH-index to provide multi-modal data indexing extension to Lucene, with performance enhancements. López et al. [21], J. A. H., & Cuadrado, J. S. (2022) developed MAR—an incremental indexed model search engine that has a user-centric interface—in total processed about 600,000 models in order to enhance precision.

TABLE I
RESEARCH PAPER STUDY

Author Name	Year	Proposed Concept	Major Findings
Kanev, A. I., & Terekhov, V. I.	2020	Evolving domain of semantic search and its relevance in information retrieval systems	Comparative analysis between semantic search and Lucene-based search, methodology for comparing search engine rankings
Kasmani, F., et al.	2020	Transition from subject-oriented to topic-oriented learning; Content-Based Search Engine for E-Books	Indexing E-books and organizing search results based on frequently occurring phrases
Jin, D., et al.	2020	Lightweight full-database retrieval approach using JDBC technology for content extraction and Lucene for full-text indexing	Efficiency and stability in retrieving extensive data from relational databases
Lin, J., et al.	2020	Interoperability among open-source search engines; Common Index File Format (CIFF)	Methods for sharing indexes between search engines, recommendation of CIFF for fair comparisons
Wang, D.	2021	Digital archive management system leveraging advanced technologies for cross-platform compatibility	Introduction of a digital archive management system leveraging advanced technologies
Hema, R.	2021	Domain-specific search application tailored for efficient retrieval of chemical documents	Specialized indexing and re-ranking based on chemical entities and phrases for enhanced accuracy
Chan, H. P., et al.	2021	Cloud-based search engine system for rich text	High recall rate and throughput in search

		content; Utilization of Apache Tika and BM25 algorithm	performance
Safaei, A.	2021	Text-based multidimensional medical image indexing technique for semantic indexing	Improvement in retrieval process in medical image search engines using feature correlations
Zouaoui, S., & Rezeg, K.	2021	Semantic search engine for the Quran based on ontology as an index	Heightened precision and recall compared to existing search engines through incorporation of semantic relations
Haddela, P., et al.	2021	Investigation of document classification methods for Sinhalese datasets	Promising results in accuracy and interpretability for document classification using evolved Lucene search queries
Azizan, A., et al.	2022	Development of a search engine using Apache Lucene framework for Malay Quran translation text	Emphasis on Lucene understanding for beginners, achieved moderate precision and recall rates
Singh, C., & Gupta, V.	2022	Introduction of a full-text search engine using Lucene; Incorporation of image search with Tesseract API	Highlighting the importance of efficient retrieval in the web environment, plans for incorporating image search
Liu, R., et al.	2022	Proposal of the MMH-index, a hybrid index structure for enhancing Lucene's capabilities	Optimization of traditional inverted indexes with modality bitmaps for improved space consumption and query performance
López, J. A. H., & Cuadrado, J. S.	2022	Presentation of MAR, a search engine for models supporting query-by-example	Enhancements include incremental indexing, keyword-based search, and user-friendly interface for improved precision

III. RESEARCH METHODOLOGY

The development of an all-rounded file search application takes a systematic methodology that covers several key stages. It all starts with the identification of the problem and its scope, and then clearly stating the need for such an application followed by specifying the functionalities that need to be included in support for the search. Requirement analysis follows to guide the choice of technologies and libraries to include, such as PyMuPDF, python-docx, Whoosh, among others, to support text extraction, along with index building processes, and consequently, search operations. Data collection and preparation take a sample dataset in various formats, process them all, and apply an indexing process with the help of the Whoosh library. The next step is to build a search algorithm with a few modes and then integrate WordNet in the process of better query expansion. Afterward, this integrates with a MySQL database for the efficient storage and retrieval of search results, along with a graphical user interface making sure there is a well-friendly interaction, developed using Tkinter. Testing and performance evaluation verify the functionality, reliability, and speed of the application, which finally leads to documentation and deployment into real-world applications, supported by full documentation and user support.

Steps :

1) **Problem Identification and Scope Definition:**

- a) Identify the need for a comprehensive file search application.
- b) Define specific functionalities required to meet user needs.
- c) Determine supported file types and scope of search functionalities.

2) **Requirement Analysis:**

- a) Conduct a thorough analysis to delineate functionalities and features.
- b) Analyze factors such as supported file formats, search modes, database integration, and GUI requirements.

3) **Technology Selection:**

- a) Select appropriate technologies and libraries for implementation.

- b) Technologies include PyMuPDF, python-docx, pytesseract, Whoosh, MySQL database, and Tkinter for GUI development.
- 4) **Data Collection and Preparation:**
 - a) Collect and prepare sample data in various file formats.
 - b) Extract text content from each file format using respective techniques.
 - 5) **Indexing Process:**
 - a) Implement indexing using the Whoosh library.
 - b) Define a schema for indexing file paths and text content.
 - 6) **Search Algorithm Development:**
 - a) Develop search algorithm to process user queries.
 - b) Implement different search modes such as direct search, similar word search, and all word forms search.
 - c) Integrate WordNet for expanding search queries.
 - 7) **Database Integration:**
 - a) Integrate MySQL database to store search results.
 - b) Design a database schema for efficient retrieval and management of search results.
 - 8) **GUI Development:**
 - a) Develop GUI using Tkinter.
 - b) Implement features like directory selection, search text entry, search mode selection, and file type selection for enhanced usability.
 - 9) **Testing and Performance Evaluation:**
 - a) Conduct rigorous testing and performance evaluation.
 - b) Evaluate indexing speed, search speed, accuracy, and gather user feedback for improvements.
 - 10) **Documentation and Deployment:**
 - a) Document implementation details including code structure, functions, and usage guidelines.
 - b) Deploy the application for real-world usage.
 - c) Provide user documentation and support for effective utilization.

IV. ALGORITHMIC MODEL

WordNet and lexical chains are two of the core constructs of computational linguistics – each of them provides a different way to explore the semantic texture of natural language. WordNet is another tool developed at Princeton University. It represents a large lexical database that classifies words into synsets and defines a web of relationships among the synsets—that is, not merely synonyms. The hierarchical structure thus permits the exploration of broader and narrower concepts that have been very facilitating to be automated in corresponding tasks. On the other hand, lexical chains are a methodological approach directed toward the unraveling of the semantic relationship in a unit of text, forming a cohesive thread of meaning based on the semantic similarity of content units. With WordNet driving and analysis, lexical chains capture inferences and bring out fine nuance holistically in text. This adds richness to many natural language processing applications, ranging from information retrieval to document summarization. WordNet and lexical chains together play the role of tools to decode meaning and richness being represented by text datasets that form the basal materials for natural language understanding and computational linguistic developments.

ALGORITHM FOR PROPOSED APPROACH

Text Extraction Module

Algorithm 1 Extract Text from PDF Files

```

1: function EXTRACT_TEXT_FROM_PDF(pdf.file.path)
2:   OPEN pdf.file.path using PyMuPDF
3:   text ← ""
4:   for each page in pdf.document do
5:     page.text ← EXTRACT_TEXT_FROM_PAGE(page)
6:     text ← text + page.text
7:   end for
8:   CLOSE pdf.file
9:   return text
10: end function

```

Algorithm 2 Extract Text from DOCX Files

```

1: function EXTRACT_TEXT_FROM_DOCX(docx.file.path)
2:   OPEN docx.file.path using python-docx
3:   text ← ""
4:   for each paragraph in docx.document do
5:     paragraph.text ← EXTRACT_TEXT_FROM_PARAGRAPH(paragraph)
6:     text ← text + paragraph.text
7:   end for
8:   CLOSE docx.file
9:   return text
10: end function

```

Algorithm 3 Extract Text from Images

```

1: function EXTRACT_TEXT_FROM_IMAGE(image.file.path)
2:   OPEN image.file.path
3:   ocr.result ← PERFORM_OCR_ON_IMAGE(image)
4:   text ← EXTRACT_TEXT_FROM_OCR_RESULT(ocr.result)
5:   CLOSE image.file
6:   return text
7: end function

```

Algorithm 4 Search Algorithm

```

1: function SEARCH(user.query, search.mode)
2:   Parse the user's search query and determine the search mode
3:   if search mode is "direct" then
4:     Construct a direct search query using the parsed search query
5:   else if search mode is "similar" then
6:     Expand the search query to include similar word forms using Word-
       Net
7:     Construct a search query with the expanded terms
8:   else if search mode is "all forms" then
9:     Expand the search query to include all word forms related to the
       given term using WordNet
10:    Construct a search query with the expanded terms
11:   end if
12:   Utilize the Whoosh library to perform the search operation on the in-
       dexed content
13:   Retrieve the matched files along with relevant information (e.g., file
       name, matched text content)
14:   return the matched files to the caller
15: end function

```

TABLE 1.

SEARCH RESULT FOR KEYWORD CAR USING BASE LUCENE SEARCH

File Name	Words Matched	Total Words
demo.txt	12	148
PFRO.txt	0	434
setupact.txt	0	751
setuperr.txt	0	0

SQL String Functions with Syntax and Examples.docx	0	1474
SVM Sentiment.pdf	0	722

The results of search operation are given in the below paper for the lucene search base for the keyword "car" across several files

The search is. It has examined various files, and the findings obtained are summarized below in the table. Of that, "demo.txt" the keyword appeared 12 times for the count of 148 words. However, it did not find the keyword in the rest of the files: "PFR0.txt," "setupact.txt," "setuperr.txt," "SQL String Functions with Syntax and Examples.docx," and "SVM Sentiment.pdf." The table gives details on how efficient or weak a Lucene search base is in fetching data of relevance to the keyword specified, and in this case "demo.txt" was the only file with matches.

TABLE 2.
SEARCH RESULT FOR KEYWORD CAR USING PROPOSED DIRECT SEARCH

File Name	Total Words	Words Matched (car)
dataimg.png	116	7
demo.txt	148	12

The table provides the result of a lines' search using a proposed algorithm as a keyword "car." Two data files were checked for this approach: dataimg.png and demo.txt. There were seven matches for the keyword "car" in 116 words under dataimg.png. Similarly, there were 12 keyword matches for "car" in 148 words under the file named "demo.txt." This table shows how the proposed direct search technique is efficient and effective in finding any keyword occurrence in similarity words with exactly the same words in different types of files with different word lengths.

Table 3
Matched Results for Keyword "car" in exactly similar words

File Name	Total Words Matched	Total Words
dataimg.png	10	116
demo.txt	17	148
SVM Sentiment.pdf	1	722

In Table 3, the matched results are displayed with the keyword "car" with exactly similar words. It compares three files: "dataimg.png," "demo.txt," and "SVM Sentiment.pdf." In the first backdrop, against 116 words in total, there were 10 instances of the keyword "car." The second file, "demo.txt," contained 17 matches for the keyword, against 148 words available. In the case of "SVM Sentiment.pdf," one exact match for the keyword "car" against a total of 722 words was observed. The table presents where the keyword 'car' is found in exactly the same forms between different files and information about its frequency in all documents.

The proposed work has some improvements over the base search strategy and provides an efficient and more accurate search compared to the base strategy. Table 1: Base Lucene search of keyword 'car' between various files with varying success. However, the direct search method did better to place the keyword occurrences correctly in "dataimg.png" and "demo.txt" files, as depicted in Table 2. Further, to eliminate the dissimilar word matching occurrence, Table 3 demonstrates the existing word matching occurrence with the keyword "car" to enhance more precisions. More general than the base approach which exists, the proposed work encountered greater efficiency of retrieving relevant information, as shown by the refinement of search capabilities realized in the tables of improved match results.

V. CONCLUSION

This paper presents a new approach to optimizing information retrieval in Lucene by integrating lexical chain analysis and WordNet.

The proposed technique thus allows for a more accurate and relevant set of search results by leveraging the semantic richness of WordNet and insight lexical chains can provide in the context of the unstructured data. It also serves as the framework supporting various search modalities while great attention is paid to keyword frequency and the presence of semantically relevant keywords. The integration of lexical chain analysis into WordNet-driven semantic analysis in a comprehensive methodology and algorithms described in the current paper holds great promise toward further enhancing information retrieval from huge and complex datasets. The research in context has been designed to address the needs of diverse domains and to extend Information Retrieval to facilitate intuitive access to Knowledge Repositories.

REFERENCES

- [1] Teofili, T., & Lin, J. (2019). Lucene for approximate nearest-neighbors search on arbitrary dense vectors. *arXiv preprint arXiv:1910.10208*.
- [2] Yilmaz, Z. A., Wang, S., Yang, W., Zhang, H., & Lin, J. (2019, November). Applying BERT to document retrieval with birch. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP): System Demonstrations* (pp. 19-24).
- [3] Liu, Q., & Xia, Z. (2019). Statement Generation Based on Big Data for Keyword Search. In *Machine Learning and Intelligent Communications: 4th International Conference, MLICOM 2019, Nanjing, China, August 24–25, 2019, Proceedings 4* (pp. 477-488). Springer International Publishing.
- [4] Madi, N., Al-Mutlaq, N., & Al-Khalifa, H. S. (2019, May). HealthSEA: Towards Improving the Search Engine of KAAHE Arabic Health Encyclopedia. In *2019 2nd International Conference on Computer Applications & Information Security (ICCAIS)* (pp. 1-7). IEEE.
- [5] Yusuf, N. U. H. U., Yunus, M. A. M., Wahid, N. O. R. F. A. R. A. D. I. L. L. A., Nawi, N. M., Samsudin, N. A., & Arbaiy, N. U. R. E. I. Z. E. (2020). Query expansion method for quran search using semantic search and lucene ranking. *J Eng Sci Technol*, 15(1), 675-692.
- [6] Ji, W. (2020, June). Research and Application of Information Data Retrieval System in Station Based on Lucene Technology. In *2020 IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA)* (pp. 687-690). IEEE.
- [7] Youzhuo, Z., Yu, F., Ruifeng, Z., Shuqing, H., & Yi, W. (2020, May). Research on lucene based full-text query search service for smart distribution system. In *2020 3rd international conference on artificial intelligence and big data (ICAIBD)* (pp. 338-341). IEEE.
- [8] Kanev, A. I., & Terekhov, V. I. (2020, December). Evaluation issues of query result ranking for semantic search. In *Journal of Physics: Conference Series* (Vol. 1694, No. 1, p. 012004). IOP Publishing.
- [9] Kasmani, F., Maniyar, R., & Narvekar, M. (2020, March). Content Based Search Engine for E-Books. In *2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS)* (pp. 528-533). IEEE.
- [10] Jin, D., Chen, G., Hao, W., & Bin, L. (2020, June). Whole database retrieval method of general relational database based on lucene. In *2020 IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA)* (pp. 1277-1279). IEEE.
- [11] Lin, J., Mackenzie, J., Kamphuis, C., Macdonald, C., Mallia, A., Siedlaczek, M., ... & de Vries, A. (2020, July). Supporting interoperability between open-source search engines with the common index file format. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 2149-2152).
- [12] Wang, D. (2021, November). Digital archive management based on lucene full-text search engine. In *Journal of Physics: Conference Series* (Vol. 2074, No. 1, p. 012001). IOP Publishing.
- [13] Hema, R. (2021). Extraction and Search of Relevant Chemical Documents from the Web. In *Proceedings of First International Conference on Mathematical Modeling and Computational Science: ICMMCS 2020* (pp. 251-262). Springer Singapore.
- [14] Chan, H. P., Xu, L., Liu, H. H., Zhang, R. T., & Sangaiah, A. K. (2021). System design of cloud search engine based on rich text content. *Mobile Networks and Applications*, 26(1), 459-472.
- [15] Safaei, A. (2021). Text-based multi-dimensional medical images retrieval according to the features-usage correlation. *Medical & Biological Engineering & Computing*, 59(10), 1993-2017.
- [16] Zouaoui, S., & Rezeg, K. (2021). A novel quranic search engine using an ontology-based semantic indexing. *Arabian Journal for Science and Engineering*, 46, 3653-3674.

- [17] Haddela, P., Hirsch, L., Brunson, T., & Gaudoin, J. (2021). Use of interpretable evolved search query classifiers for sinhala documents. In *Proceedings of the Future Technologies Conference (FTC) 2020, Volume 1* (pp. 790-804). Springer International Publishing.
- [18] Virginia, G., Tjandra, J. C., Susanto, B., & Proboyekti, U. (2021, November). Search engine optimization by re-ranking the product search result based on user click data. In *Proceedings of the 3rd International Conference on Advanced Information Science and System* (pp. 1-5).
- [19] Azizan, A., Mohd Sanusi, N. I. N., Khairuddin, N., & Shafie, A. S. (2022). Lucene Search Engine Development: a beginner's experience. *Mathematical Sciences and Informatics Journal (MIJ)*, 3(2), 80-92.
- [20] Singh, C., & Gupta, V. (2022). Framework for Faster Index Search. In *Rising Threats in Expert Applications and Solutions: Proceedings of FICR-TEAS 2022* (pp. 439-446). Singapore: Springer Nature Singapore.
- [21] Liu, R., Liang, J., Jin, P., & Wang, Y. (2022, October). MMH-index: Enhancing Apache Lucene with High-Performance Multi-Modal Indexing and Searching. In *Proceedings of the 30th ACM International Conference on Multimedia* (pp. 7279-7289).
- [22] López, J. A. H., & Cuadrado, J. S. (2022). An efficient and scalable search engine for models. *Software and Systems Modeling*, 21(5), 1715-1737.
- [23] Lin, J., Pradeep, R., Teofili, T., & Xian, J. (2023). Vector search with OpenAI embeddings: Lucene is all you need. *arXiv preprint arXiv:2308.14963*.
- [24] Scells, H., & Potthast, M. (2023, July). pybool_ir: A Toolkit for Domain-Specific Search Experiments. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 3190-3194).
- [25] Chen, H., Lassance, C., & Lin, J. (2023). End-to-end retrieval with learned dense and sparse representations using Lucene. *arXiv preprint arXiv:2311.18503*.
- [26] SKA, Manish Kumar Mukhija, and Pooja Singh "A Security Approach to Manage a Smart City's Image Data on Cloud," *AI-Centric Smart City Ecosystems: Technologies, Design and Implementation* (1st ed.), PP: 68-82, (2022). CRC Press. <https://doi.org/10.1201/9781003252542>
- [27] S. K. "A. Raj, V. Sharma, and V. Kumar." "Simulation and Analysis of Hand Gesture Recognition for Indian Sign Language Using CNN". *International Journal on Recent and Innovation Trends in Computing and Communication* 10, no. 4 (2022): 10-14.
- [28] SKA "An Improved Algorithm for Faster Multi Keyword Search in Structured Organization", *International Journal on Future Revolution in Computer Science & Communication Engineering* Vol-5, issue- 5 (2019), 19–23.
- [29] SKA, "A New Palm Print Recognition Approach by Using PCA & Gabor Filter", *International Journal on Future Revolution in Computer Science & Communication Engineering*, Vol-4, Issue-4 (2018), 38–45.
- [30] S. K. . Analysis of WAF and Its Contribution to Improve Security of Various Web Applications: Benefits, Challenges. *ijfrcsce* 2019, 5, 01-03.