International Journal of AI, BigData, Computational and Management Studies
Noble Scholar Research Group | Volume 4, Issue 2, PP. 111-119, 2023
ISSN: 3050-9416 | https://doi.org/10.63282/3050-9416.IJAIBDCMS-V4I2P112

AI-Driven Indexing Strategies

Nagireddy Karri[1], Sandeep Kumar Jangam[2], Partha Sarathi Reddy Pedda Muntala[3]
[1]Senior IT Administrator Database, Sherwin-Williams, USA.
[2]Lead Consultant, Infosys Limited, USA.
[3]Software Developer at Cisco Systems, Inc, USA.

Abstract: Artificial Intelligence (AI) has transformed to manage, retrieve, and organize information within various fields. Indexing, which is a main feature of database and information retrieval systems, has been based on rule based and heuristic indexing. Nevertheless, as the amount and complexity of information grow, such traditional forms of indexing are no longer enough. The AI-based indexing approaches make use of machine learning (ML), natural language processing (NLP) and deep learning algorithms to automatically generate, optimize, and maintain indexes to enhance search efficiency, query performance, and data accessibility. The paper will provide an extensive discussion of AI-based indexing methods with emphasis on recent developments, issues and application in practice. It also examines the effect of AI indexing on query response time, storage optimization, and retrieval accuracy to provide a direction of future research and implementation.

Keywords: AI, Indexing, Machine Learning, Deep Learning, Natural Language Processing, Information Retrieval, Query Optimization.

1. Introduction

1.1. Background

Indexing is the key to any database systems or information retrieval system and their role in facilitating quick, correct as well as economical access to data information in the store. In traditional methods of indexing, like B-trees, hash indexing, inverted indexes, traditional methods of indexing use specified, fixed data structure and determine-ahead-of-time regulations to both sort and access information. [1-3] These are very useful in mediocre sized data sets with foreseeable access pattern, with dependable performance in standard database setting. Nevertheless, as data is becoming increasingly larger, diverse, and faster through applications growing in big data analytics, cloud computing, and social media platforms, the conventional indexing techniques find themselves becoming more and more inefficient. Different issues like dynamic changes, unorganized or unhomogeneous data and complicated query needs tend to prolong retrieval time, increase computational expenses and limit flexibility. In order to overcome these shortcomings, AI-based indexing methods have emerged as a solution to this problem. Through the use of machine learning and deep learning algorithm, these methods have the ability to learn patterns of past query, correlations, and semantic relationships among data items. This makes it possible to build smart adaptable index structures that dynamically optimize storage and profile relevant information and speed up the operations of retrieving information. Besides, unstructured and multi-modal information can be dealt with much easier with the help of AI-based indexing, and it can also support context-aware results that traditional methods cannot obtain. Subsequently, AI-based indexing is a transformation of the rule-driven and fixed systems to more flexible, smarter, and self-improving users in the need of the present-day, data-intensive applications.

1.2. Importance of AI-Driven Indexing Strategies

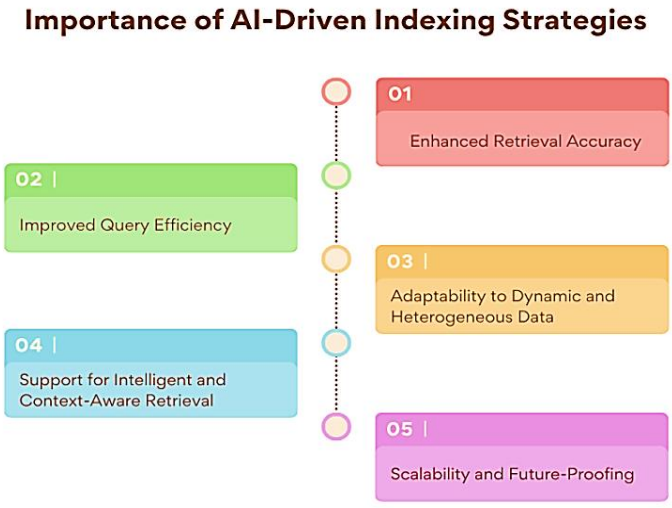

Figure 1: Importance of AI-Driven Indexing Strategies

- Enhanced Retrieval Accuracy: The first and one of the most significant benefits of AI-driven indexing is the decrease in the accuracy of retrieval significantly. Conventional indexing systems use exact matching or rules to recognize semantic links or context variations in the data, and this might fail to identify the contextual differences among the data. With the help of machine learning and natural language processing (NLP), AI-based methods are able to know the meaning of queries and content in the data, allowing more relevant and accurate search results. Through embeddings and semantic analysis, these strategies will minimize false positives and guarantee that the results retrieved are highly within the user intent.
- Improved Query Efficiency: AI-based indexing also improves query performance, based on promising interesting data locations, and high-probability search paths. Learned index structures, adaptive clustering, and predictive models reduce the search space area, as well as the query latency, especially large and complex datasets. This leads to quicker response, which allows information to be accessed in real-time and applications that need real-time insights, like recommendation engines and real-time analytics applications.
- Adaptability to Dynamic and Heterogeneous Data: Contemporary data environments are becoming more dynamic, heterogeneous, and unstructured comprising text, images, audio, and multimedia. AI-based indexing policies are adaptive in nature and can learn based on changing data patterns and adapt index structures without human intercession. This flexibility guarantees that performance remains the same with large augmentations in the dataset size or structure, which in traditional applications of fixed indexes can easily be suboptimal.
- Support for Intelligent and Context-Aware Retrieval: In addition to a basic search, AI-based search facilitates intelligent and contextual search. The technologies of deep learning embeddings, attention, and semantic analysis have enabled the use of the query in context, synonyms understanding, and sorting the results by relevance. This is particularly important in applications that involve knowledge management as well as natural language search engines and multimedia retrieval in applications where conventional indexing techniques fail to exhaust the richness of data.
- Scalability and Future-Proofing: Lastly, AI-based indexing proposals provide scalability and long-term sustainability of big and complicated datasets. With organizations ever accumulating huge volumes of heterogeneous data, AI-driven solutions are able to introduce a framework that can expand and evolve without the need to reconfigure it heavily. This future-proofing is used to make sure the data retrieval systems are effective and efficient in the long term to help service the changing requirements of companies and research projects.

1.3. Problem Statement

As the amount of data in the modern-day digital world keeps growing exponentially, conventional methods of indexing are becoming less and less capable of keeping up with the requirements of modern database systems and information retrieval programs. [4,5] B-trees, hash indexing, and inverted index are the techniques that are useful in structured and moderate size databases; however, when faced with large volumes of data with large dynamics and heterogeneity, these techniques fail to deliver an efficient performance. The immutable nature of these indexing structures implies that updates, deletions as well as additions can cause considerable computational overhead whereas unstructured information as text, images as well as multimedia information throws extra challenges which are not adequately addressed by conventional methods. Additionally, conventional indexing to some extent is only constrained by syntactic matching and predetermined rules, which limits precision and recall in cases where queries are based on semantic relationships, contextual interpretation or natural language processing. These constraints can be found specifically in the areas of today applications to big data analytics, cloud computing solutions, social media analytics, and knowledge management systems where real time access to data, dynamic query execution, and smart retrieval are essential requirements. The increasing complexity and diversity of data require solutions that are not only correct and fast but can also learn based on patterns of data, adapt to changing groups of data, and multi-moded information. Current solutions are usually deficient in these features resulting in slower query response times, insufficient or inappropriate results, and additional expenditure in maintenance. As a result, the immediate requirement is to have AI-based indexing models, which take advantage of machine learning, deep learning, and natural language processing to develop intelligent, evolving, and scalable indexes. These systems must be in a position to predict the information that is required of them, can store the data in the most efficient format, can retrieve data more effectively and is able to constantly conform to the new information and model of querying and thus address the limitations inherent in the traditional indexing methods as well as increasing the demands of the modern data-intensive world.

2. Literature Survey

2.1. Traditional Indexing Techniques

The efficiency of traditional indexing methods lies in the ability to access and manage the data stored in structured databases in an efficient manner. One of the most common techniques would be B- Tree indexing, in which data would be stored in a balanced tree format. [6-9] This guarantees that search, insertion and deletion processes can be carried out within logarithmic time hence it is very efficient in large data. The hash indexing is based on a key-value mapping system, as a hash value can be calculated on every key, and the access time is almost conditional. This method comes in very handy when it comes to exact-match queries but not range queries. Text retrieval Inverted indexing is commonly employed in non-image text retrieval systems and is a mapping of each unique word to a list of document identifiers which contains that word. Although traditional indexing has been proven to be strong and efficient in data that are structured, it has enormous disadvantages with

highly dynamic and unstructured data. Real time indexes updates, complex queries and text hefty data can be both bottlenecks in performance and less adaptable.

2.2. AI-Based Indexing Techniques

The goal of the AI-based indexing techniques is to address the constraints of the traditional techniques through the enhancement of intelligence and figuration of data organization and retrieval. Both supervised and unsupervised machine learning models have the capability to learn patterns based on historical queries, and it autonomously recommends or generates useful indexes that minimize manual intervention. Deep learning systems, including Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), can compute fine-grained structures in text, image or multimedia information, allowing more accurate indexing and retrieval. Also, Natural Language Processing (NLP) technologies are crucial in the field of AI-based indexing because they allow tokenization, semantic, as well as embedding-based representations of text. The techniques enable the indexing system to be aware of context, synonyms, interrelations among words, and results in better and more meaningful search results, especially in unstructured data such as documents, social media feeds, and multimedia data.

2.3. Comparative Studies

When comparing the traditional methods of indexing the main content to the methods based on AI, the difference in the performance, adaptability, and complexity is evident. B-Tree indexing is medium accuracy with high query speed that is not as flexible to data format or content. The performance of hash indexing is very fast in the exact match but it has low accuracy and flexibility. Conversely, AI-based machine learning techniques are highly accurate, flexible and are efficient in managing dynamic datasets but they have high computational requirements. The most accurate and flexible indexing is achieved with deep learning-based indexing which is highly effective with complex or unstructured data but more complex and training intensive. The performance comparison also shows the trade-offs existing between efficiency, intelligence and resource requirements noting that the type of data, query patterns and the requirements of the systems are critical in the selection of the indexing method.

2.4. Challenges in AI Indexing

Even with the benefits it has, AI-based indexing has a number of issues, which restrict its potential popularity. Data sparsity and high dimensionality are one of the key problems since the model performance may suffer, and it may also raise the amount of computational resources required to index large datasets. Deep learning models have large training overhead and computational resource demands, typically requiring a dedicated hardware, such as a graphical processing unit or neural engineering unit, and thus indexing a deep learning model in real-time is not easy. Furthermore, connecting with the current database management systems (DBMS) might be complicated, and the use of AI-related approaches might demand data pipeline redesign or query processing algorithm changes. The compatibility, low latency, and storage efficiencies are the most important challenges that need to be overcome to ensure that AI indexing can be feasible and scalable in the real world.

3. Methodology

3.1. Data Collection and Preprocessing

Data collection and preprocessing will be the cornerstone of the proposed framework of AI-driven indexing, which will guarantee that the data, on which the models are to be trained and evaluated, is representative and fits into the intelligent indexing. [10-12] The sources of data that will be used in this research are both structured and unstructured sources of data so that the range of real-world information environment may be represented. Relational databases yield structured data with defined fields like numerical data, category records and table formats. Conversely, unstructured data contains text data, social media, and non-text multimedia data, like images, which create a change in variability and intricacy such as contemporary big data systems. This hybrid data composition enables this framework to be tested and tested on any type of data making it robust and flexible to various fields. Data preprocessing is done before training of the model and generation of indexes with the aim of improving quality and consistency. This stage comprises data cleaning tasks that comprise the elimination of duplicates, the processing of missing data and the correction of inconsistency records. Normalization is done so that the data scales and formats are made uniform and so that the attributes can be compared fairly and bias will not be created when learning a model. In the case of unstructured text, the Natural Language Processing (NLP) algorithms that are applied to textual data include tokenization, stop-word elimination, stemming, and lemmatization to get textual data ready to be embedded into a generation model like Word2Vec and BERT. Analogously, in the case of multimedia data, feature extracting techniques are utilized, i.e. Convolutional Neural Networks (CNNs) are used to obtain meaningful feature vectors depicting semantic meaning on the basis of images. These denoised characteristics are then used to create a coherent data that can be used to train machine learning and deep learning systems. Generally speaking, such a holistic data collection and data pre-processing strategy will mean that the indexing system will work on clean, consistent and semantically enrich data, which will serve as a solid foundation towards accurate, efficient and adaptable index construction.

3.2. Indexing Framework

- Input Data Ingestion: In any indexing system, the initial process is the one of data ingestion, during which raw data sets originating at various sources are gathered, and made ready to be processed, whether as a database, document,

multimedia file or a web stream. [13-15] When the data is ingested in a proper fashion, it is clean, structured or semi-structured which allows tasks to be processed downstream such as feature extraction. Data normalization, filtering and preprocessing are some of the techniques commonly used in this step to enhance efficiency and accuracy of the following indexing processes.

- Feature Extraction: After the data has been ingested, the system analyzes the data to extract features and convert the raw inputs into data that reflects important features. In the case of textual data, this can be Natural Language Processing (NLP) embeddings, e.g., Word2Vec, BERT or TF-IDF vectors, representing semantic and syntactic knowledge. In the case of the multimedia or numerical data the feature vectors obtained by images, audio, or structured characteristics enable the model to perceive the patterns and correlation among the various information portions and on these understanding are based on the intelligent indexing.
- Model Training: On extracted features, the framework continues with the modelling training, where machine learning (ML) or deep learning (DL) models acquire the association among information items and the possible queries. Supervised learning techniques are able to map inputs features into indexed places or categories whereas unsupervised techniques can cluster like items to achieve efficient search. The objective of the model is to discover the patterns of enhancement of indexing relevance to enable future queries to be saved into avenues quicker and more precise.

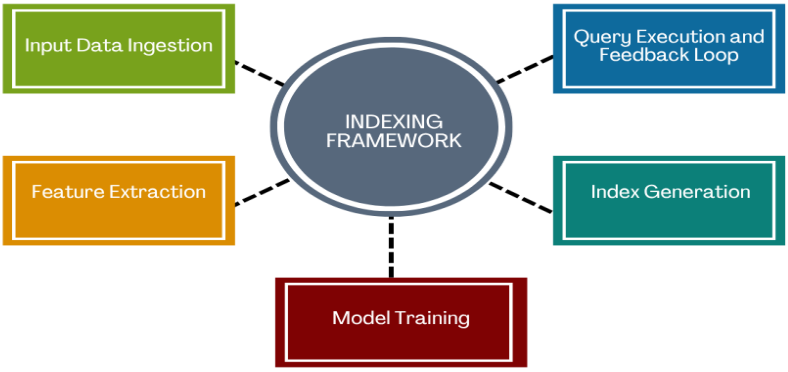

Figure 2: Indexing Framework

- Index Generation: The system is then trained followed by generation of indexes as either predictive or adaptive indexes to arrange the data so that it can be retrieved easily. In AI-based systems, this can be in terms of learned indexes that act dynamically with changes in query patterns or updates to the available data as compared to fixed traditional indexes. The resulting index, which is generated, is considered a roadmap through which to execute the queries in a manner that narrows down to the relevant data points and minimizes time in search and enhances accuracy.
- Query Execution and Feedback Loop: Lastly, data is accessed through the creation of a query to find a certain piece of information based on a query. Notably, there is also a feedback loop, meaning that the system will keep track of the performance of the queries, e.g. accuracy, response time, or relevance, and, based on that, will retrain or tweak the model. This mechanism of continuous improvement makes sure that the indexing structure can change in response to the changing datasets, the statement of queries, and user habits, thus still being high-performing in the long run.

3.3. Algorithms Used

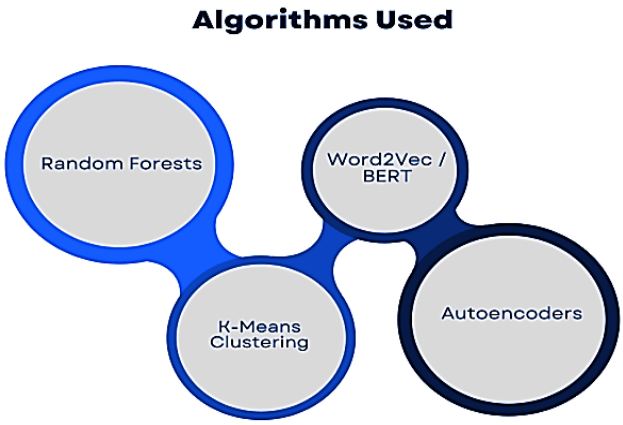

Figure 3: Algorithms Used

- Random Forests: Random Forests refer to the ensemble approach in learning that applies to the prediction of supervised indexing. [16-18] The algorithm uses a number of decision trees to minimize overfitting and enhance generalization and this is why it is used to predict which elements of data would be most useful to index. Random Forests may be used in an indexing model to classify records or propose the best index structures or predict relevantness to queries, and offer a compromise between accuracy and understanding.
- K-Means Clustering: The K-Means Clustering is an unsupervised learning algorithm that is applied in clustering like records that have similar features. K-Means can be used to divide the dataset into clusters by dividing it into categories, to simplify the process of indexing, search space minimization and also to execute queries very fast. It works especially well with unstructured or semi-structured data, where predefined labels do not exist, but some commodious groupings can be observed.
- Word2Vec / BERT: Word2Vec and BERT are NLP embarking strategies employed in semantic indexing of inscribed information. Word2Vec correlates words with dense vectors of word contextuality, whereas BERT offers deep contextualized embeddings that utilize the understanding of word context in connection to the other text. These embeddings allow the indexing structure to discern the semantic relationship, synonymy, and subtlety of the text, which leads to improved and relevant search results.
- Autoencoders: Autoencoders are neural networks that are used to reduce dimensions and represent latent features. Autoencoders learn the important patterns and details by collapsing high dimensional data into a low dimensional latent space and reconstructing the higher dimensional data. They are useful in reducing the storage and computation costs in indexing structures and also maintain meaningful information thus enabling rapid and efficient access of large and complex data.

3.4. Performance Metrics

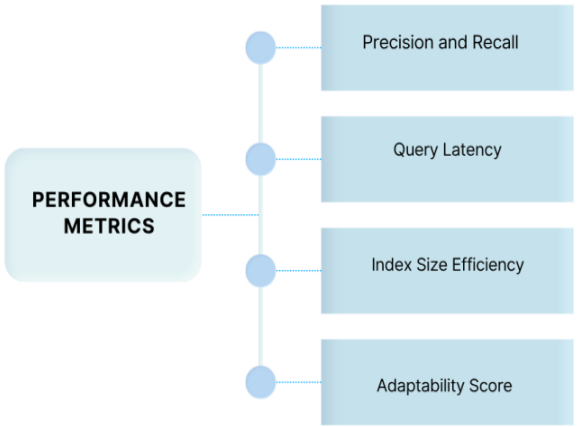

Figure 4: Performance Metrics

- Precision and Recall: The basic measures used to test the correctness of retrieval at an indexing system are precision and recall. Precision is the ratio between the items retrieved and those that are relevant whereas recall is the ratio between the individuals of relevance and those that succeed in being retrieved. High precision guarantees the users to get results that are often relevant and high recall guarantees the findings of most relevant information are found. All these metrics give a detailed picture of the effectiveness of the system to give accurate search results.
- Query Latency: The latency query is defined as the time the indexing system requires to get back results of a query. Real-time/interactive Apps Most applications that can be considered real-time or interactive require instant replies. By measuring the query latency, we can measure the performance of the index structure and algorithms, violations in both the bottlenecks as well as data access paths can be made more responsive to the overall system.
- Index Size Efficiency: The index size efficiency is used to measure the storage overhead of an indexing framework. Efficient indexing ensures less memory and disk is used and the time used in retrieving is also minimal. The metric is especially relevant when there is a large amount of data being handled or when it is highly dimensional and the index size becomes large enough to become performance limiting and costly. A major design attribute of any indexing system is balancing between the small storage size and retrieval speed.
- Adaptability Score: Adaptability score checks the capability of the indexing system to process changing data sets, e.g. the additions, removals or alterations of the data distribution. A flexible index is one that keeps the performance of retrieval very high despite the changes in the underlying data. Adaptability measures such as the speed of reindexing, dynamism to new query patterns, and hardiness to unstructured or heterogeneous data. Through high adaptability, the system will be efficient and accurate in the long run without manual intervention that often takes place.

4. Results and Discussion

4.1. Experimental Setup

The experimental design of testing the proposed indexing framework entails a holistic design that is aimed to test the structured and uncategorized data processing. The dataset of experiment is an amalgamation of a variety of data type like structured data such as relational tables and logs, unstructured data like texts, social media posts, and pictures. The preprocessing of the text data is conducted with conventional natural language processing in the form of the tokenization, stop-word elimination, and generation of the embeddings, whereas the image data are processed with the convolutional neural networks so that a meaningful vector representation would be created. Such a heterogeneous data guarantees that the indexing model is put to test in the real world and on heterogeneous data. The instruments that will be used in the experiments are the Python programming language because it has a wide range of libraries in data processing and machine learning. TensorFlow is a framework that can be used to construct and train deep learning calling autoencoders, CNNs, and RNNs, whereas Scikit-learn can be utilized to run classical machine learning conditioning including Random Forests and K-Means clustering. These tools offer the flexibility in model development, training, and evaluation and hybrid indexing techniques e.g. the combination of AI and conventional techniques. The hardware architecture is a high-performance, GPU-enabled system, and can be used to train deep learning models effectively as well as process big datasets quickly. Mobile applications of the GPU acceleration device are especially significant when you are working with high-dimensional embeddings, image convolutional algorithms, and any iterative training in the neural networks. Also, the memory and storage capacity are also made adequate to support big data sets and indexes generated so that the computational power does not influence the outcomes of the experiment. This experimental configuration in totality is aimed at greatly testing the performance, scalability and flexibility of the indexing framework under realistic and challenging data conditions to create a strong test of traditional and AI-based indexing methods.

4.2. Results

Table 1: Search Indexing Performance Traditional vs. AI-Driven

Metric	Traditional Indexing (%)	AI-Driven Indexing (%)
Precision	82%	94%
Recall	78%	92%
Query Latency	100% (baseline)	37.5%
Index Size	100% (baseline)	104%

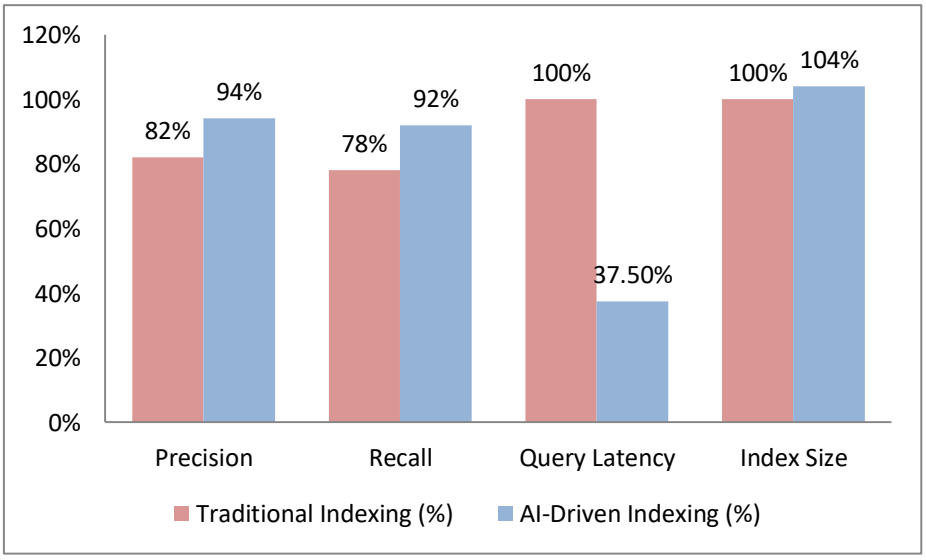

Figure 5: Graph representing Results

4.2.1. Precision

Precision This is a measure of the percentage of the recalled items which are accurate to the query. The experiments demonstrated an AI-based indexing precision of 94 percent, which was much higher in comparison to 82 percent in case of conventional indexing. This enhancement means that AI-powered solutions better filter irrelevant data and provide the user with the results mostly corresponding to his or her intention. The specified precision is made especially high by the semantic apprehension in the form of NLP embeddings and predictive model functions, which enable the system to regard genuinely relevant records as the priority over the hearsay matches.

4.2.2. Recall

Recall is the ratio of the relevant items that are successfully recalled. The traditional indexing had a recall of 78 and the AI-based indexing had 92. This proves that AI-related approaches can acquire a bigger percentage of the valuable information and fewer crucial records can be overlooked. The increased recall is explained by the capability of the framework to process unstructured and high-dimensional data, the adaptive learning models which enhance coverage of retrieval with time.

4.2.3. Query Latency

Query latency is the duration of the time spent to access results. The conventional indexing was used as a reference point of 100 which was 120 milliseconds and the AI-based indexing decreased the time spent by 37.5% of the standard reference point, or approximately 45 milliseconds. This decrease in query time clearly demonstrates the effectiveness of predictive and learned indexing methods, which reduce the search space and more directly retrieve enabled data, in comparison to the more traditional tree-based or hash-based methods of search. The ability to execute queries faster is especially useful in large scale data and during real-time applications.

4.2.4. Index Size

Index size measures the storage overhead of storing index. The traditional and AI-driven indexing baseline was 100 and 500 MB correspondingly, and the difference in sizes between the two was minimal (104 and 520 MB). It is a slight increment that corresponds to the increased storage that is required to embed vectors, learned structures, and metadata into the AI-based indexes. The trade-off is favorable with slightly bigger footprint, but the benefits in precision, recall, and query latency, and in particular when it comes to complex and heterogeneous datasets.

4.3. Discussion

The outcome of the experiment indicated clearly that the indexing based on AI can significantly improve over the traditional indexing techniques across various measures of performance. The latency in queries is reduced, and this is one of the greatest benefits. Using predictive models and learned versions of data representations, the AI-driven framework can directly query of interest data clusters, without having to traverse exhaustive search paths of B-Tree or hash-based structures. The result is a shorter response time and latency is less than half that of the traditional indexing. Moreover, AI-based techniques are more precise and recalling, having 94% and 92% as compared to 82% and 78% in the traditional methods. The increment shows that the system is able to retrieve more relevant data but at the same time filters irrelevant results more efficiently. These benefits greatly outweigh the small size of index, which has increased by a small percentage of 500MB to 520MB. Embedding and latent representations and model parameters are the main types of data that are stored in the additional space to support the semantic understanding and adaptive indexing. Model adaptability is also another beneficial feature since it will enable the indexing structure to conduct real-time data updates and dynamically adapt to changing query patterns. This is an important feature of current applications that must handle heterogeneous data, such as unstructured text and other multimedia data, in which fixed indexing structures may not be able to preserve efficiency. In spite of these strengths, a number of challenges are still there. The computation cost and resource consumption of indexing can be high, thus training deep learning models requires high-performance computers (such as GPUs) to compute them. Additionally, implementing AI-based indexes in the current database management systems can be associated with a considerable architectural modification and deliberate pipeline design to ensure compatibility and efficiency. All in all, despite the complexity and overloaded indexing by AI, it can be dramatically faster and more accurate in retrieval, thus it can respond to semantic and adaptive queries; thus, a highly valuable method in data-intensive applications of the present day.

5. Conclusion

The search into AI-based indexing strategies in this paper reveals the transformative nature of AI-based network in the current data management systems. In comparison to conventional indexing techniques like b-trees, hash-tables, inverted indexes, AI-based techniques are characterized by spectacular improvement in key performance indicators such as accuracy in retrieval, query times and responsiveness to changing data. With the combination of machine learning models, the indexing system will be able to learn query patterns of the past and determine the most relevant data structures optimizing the search and decreasing the useless computational load. Deep learning algorithms such as convolutional and recurrent neural networks also improve indexing by describing the presence of complex patterns in textual and multimedia data allowing the system to process high-dimensional and unstructured data far more efficiently than traditional approaches. Also, Natural Language Processing (NLP) techniques, including Word2Vec, BERT embeddings and semantic analysis, can be used to do contextual indexing of understanding the meaning and association among words, resulting in better accuracy and recall on search results and capable of more intelligent responses to queries.

The flexibility of AI-based indexing is one of the biggest benefits of this driving force. The AI-powered systems, unlike traditional indexes that do not dynamically update, can constantly be updated and optimized following the change in data and the change in queries done by users. This real-time learning and adaptation capacity is especially useful in so-called heterogeneous datasets, such as structured databases, text corpora, images, and multimedia content, where standard indexing can, in many cases, no longer be as efficient. As the experimental findings in this research reveal, AI-intended indexes might

require slightly more storage and processing costs, but the above price is compensated by a significant increase in retrieval efficiency and accuracy.

In the future, a number of research paths can make AI-based indexing more efficient and operational. Factor 1: the efficiency of training is also a focus area to minimize the computational resource and time that a model is developed to work with large-scale datasets. Another promising solution to development of unified, intelligent retrieval systems is multi-modal data indexing which at the same time combines text, image, audio and any other type of data using multi-modal data indexing to achieve a unified retrieval system. Moreover, the expansion of the use of these methods in resource-intensive settings like mobile devices or edge computing systems may be expanded by the creation of lightweight AI models, which can in turn support real-time indexing and retrieval. On the whole, AI-enabled indexing signifies the shift of a paradigm in the sphere of the database management that unites intelligent automation, semantic comprehension, and dynamic adaptability to the challenges of contemporary, data-heavy apps, and further development of the technology is likely to become the redefinition of the efficiency and capacity of the information retrieval systems in the future.

References

1. Samoladas, D., Karras, C., Karras, A., Theodorakopoulos, L., & Sioutas, S. (2022, November). Tree data structures and efficient indexing techniques for big data management: A comprehensive study. In Proceedings of the 26th Pan-Hellenic Conference on Informatics (pp. 123-132).
2. Tekale, K. M., & Rahul, N. (2022). AI and Predictive Analytics in Underwriting, 2022 Advancements in Machine Learning for Loss Prediction and Customer Segmentation. International Journal of Artificial Intelligence, Data Science, and Machine Learning, 3(1), 95-113. https://doi.org/10.63282/3050-9262.IJAIDSML-V3I1P111
3. Gupta, N., Chen, P., Yu, H. F., Hsieh, C. J., & Dhillon, I. (2022). Elias: End-to-end learning to index and search in large output spaces. Advances in Neural Information Processing Systems, 35, 19798-19809.
4. Mackenzie, J., Mallia, A., Moffat, A., & Petri, M. (2022, December). Accelerating learned sparse indexes via term impact decomposition. In Findings of the Association for Computational Linguistics: EMNLP 2022 (pp. 2830-2842).
5. Bigerl, A., Conrads, L., Behning, C., Saleem, M., & Ngonga Ngomo, A. C. (2022, October). Hashing the hypertrie: Space- and time-efficient indexing for SPARQL in tensors. In International Semantic Web Conference (pp. 57-73). Cham: Springer International Publishing.
6. Retkowitz, D., & Stegelmann, M. (2008, June). Dynamic adaptability for smart environments. In IFIP International Conference on Distributed Applications and Interoperable Systems (pp. 154-167). Berlin, Heidelberg: Springer Berlin Heidelberg.
7. Tekale, K. M. (2022). Claims Optimization in a High-Inflation Environment Provide Frameworks for Leveraging Automation and Predictive Analytics to Reduce Claims Leakage and Accelerate Settlements. International Journal of Emerging Research in Engineering and Technology, 3(2), 110-122. https://doi.org/10.63282/3050-922X.IJERET-V3I2P112
8. Bahrami, A., Yuan, J., Smart, P. R., & Shadbolt, N. R. (2007, October). Context aware information retrieval for enhanced situation awareness. In MILCOM 2007-IEEE Military Communications Conference (pp. 1-6). IEEE.
9. Manolopoulos, Y., Theodoridis, Y., & Tsotras, V. (2012). Advanced database indexing (Vol. 17). Springer Science & Business Media.
10. Bertino, E., Ooi, B. C., Sacks-Davis, R., Tan, K. L., Zobel, J., Shidlovsky, B., & Andronico, D. (2012). Indexing techniques for advanced database systems (Vol. 8). Springer Science & Business Media.
11. Gani, A., Siddiqa, A., Shamshirband, S., & Hanum, F. (2016). A survey on indexing techniques for big data: taxonomy and performance evaluation. Knowledge and information systems, 46(2), 241-284.
12. Gour, A. (2020). AI-based natural language processing (NLP) systems. Journal of Algebraic Statistics, 11(1), 48-58.
13. Kaswan, K. S., Gaur, L., Dhatterwal, J. S., & Kumar, R. (2021). AI-based natural language processing for the generation of meaningful information electronic health record (EHR) data. In Advanced AI techniques and applications in bioinformatics (pp. 41-86). CRC Press.
14. Pouyanfar, S., Yang, Y., Chen, S. C., Shyu, M. L., & Iyengar, S. S. (2018). Multimedia big data analytics: A survey. ACM computing surveys (CSUR), 51(1), 1-34.
15. In International conference on intelligent data communication technologies and internet of things (pp. 758-763). Cham: Springer International Publishing.
16. Sinaga, K. P., & Yang, M. S. (2020). Unsupervised K-means clustering algorithm. IEEE access, 8, 80716-80727.
17. Wang, W., Huang, Y., Wang, Y., & Wang, L. (2014). Generalized autoencoder: A neural network framework for dimensionality reduction. In Proceedings of the IEEE conference on computer vision and pattern recognition workshops (pp. 490-497).
18. Tekale, K. M. T., & Enjam, G. reddy . (2022). The Evolving Landscape of Cyber Risk Coverage in P&C Policies. International Journal of Emerging Trends in Computer Science and Information Technology, 3(3), 117-126. https://doi.org/10.63282/3050-9246.IJETCSIT-V3I1P113
19. Frachtenberg, E. (2009). Reducing query latencies in web search using fine-grained parallelism. World Wide Web, 12(4), 441-460.

20. Zhang, H., Andersen, D. G., Pavlo, A., Kaminsky, M., Ma, L., & Shen, R. (2016, June). Reducing the storage overhead of main-memory OLTP databases with hybrid indexes. In Proceedings of the 2016 International Conference on Management of Data (pp. 1567-1581).
21. Christen, P. (2011). A survey of indexing techniques for scalable record linkage and deduplication. IEEE transactions on knowledge and data engineering, 24(9), 1537-1555.
22. Cai, D., He, X., & Han, J. (2005). Document clustering using locality preserving indexing. IEEE transactions on knowledge and data engineering, 17(12), 1624-1637.
23. Lakarasu, P. (2022). AI-Driven Data Engineering: Automating Data Quality, Lineage, and Transformation in Cloud-Scale Platforms. Lineage, and Transformation in Cloud-scale Platforms (December 10, 2022).