WCRG Global Research
WORLD COMET RESEARCH GROUP

International Journal of Artificial Intelligence & Digital Transformation
Vol. 5, No. 2, 2022
Doi: 10.67228/30713315/IJAIDT-2022PII1W2J
PP. 01-14

Original Article

AI-Powered Intelligent Search Systems for Big Data Environments

Dr. Sibusiso

Department of Management Studies, Mbabane State University, Eswatini.

Received: 09-04-2022 Revised: 09-26-2022 Accepted: 10-02-2022 Published: 10-04-2022

ABSTRACT

Contemporary digital ecosystems generate vast and diverse data, making traditional keyword-based search engines insufficient. This paper analyzes intelligent AI-based search systems in big data environments, highlighting developments up to 2018. It explains how techniques like machine learning, natural language processing, deep learning, and semantic reasoning enhance search accuracy, scalability, and contextual understanding. These systems use learning-based ranking, user behavior analysis, and adaptive indexing to deliver personalized and efficient results, while handling unstructured data and dynamic updates. The study discusses system components such as data ingestion, preprocessing, indexing, query understanding, ranking, and feedback mechanisms. It reviews key advancements in semantic search, learning-to-rank, and distributed search systems. Performance evaluation using metrics like precision, recall, F1-score, and latency shows significant improvement over traditional methods. The findings confirm that AI-based search systems provide more accurate and user-satisfying results, especially for large-scale unstructured data. Challenges such as data privacy, computational complexity, and interpretability are also addressed, along with future directions like reinforcement learning and real-time adaptive systems.

KEYWORDS

Artificial Intelligence, Big Data, Intelligent Search Systems, Machine Learning, Natural Language Processing, Information Retrieval, Distributed Computing, Semantic Search, Deep Learning.

WCRG
Global Research
ISSN: 3071-3315
PP: 01-14.

1. INTRODUCTION

1.1. Background

The blistering development of digital technologies has led to the massive rise of data creation in a variety of spheres of life including healthcare, finance, social media, and science research. This big data, as it has often been referred to, is defined by its size, speed, and the range of information thus difficult to process using the traditional information retrieval systems and extract any meaningful information. Traditional search engines that mostly utilize key-word searching methods usually do not give a full picture of the context and motive of the user query resulting in incomplete or irrelevant search results. Along with the development of artificial intelligence, there have been new opportunities to improve the search abilities by incorporating machine learning and situational insight to retrieve more accurate, relevant, and intelligent information.

1.2. Importance of AI-Powered Intelligent Search Systems

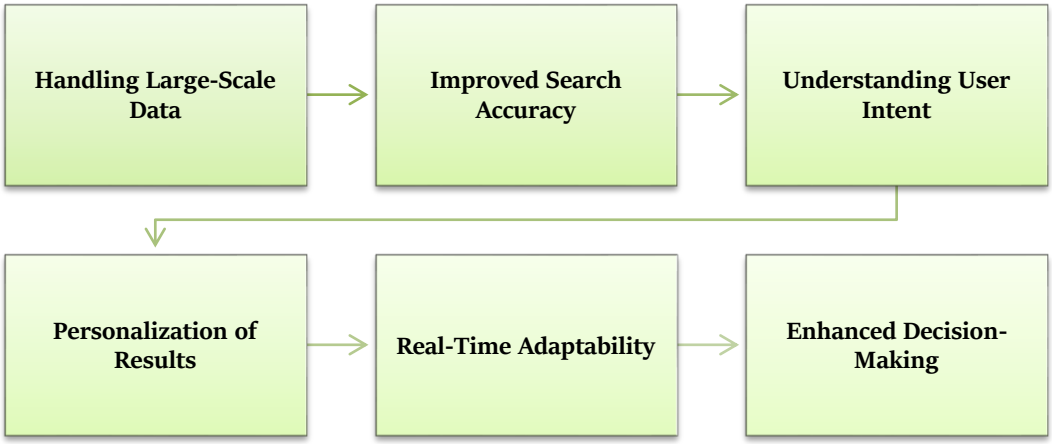

Fig 1 - Importance of AI-Powered Intelligent Search Systems

1.2.1. Handling Large-Scale Data

The search systems powered by AI are critical to the management and analysis of large volumes of data that are produced in the contemporary digital setting. They are also capable of effectively handling big data using advanced algorithms and distributed computing unlike the traditional systems, which makes them much faster and more precise in retrieving information contained in large datasets.

1.2.2. Improved Search Accuracy

The use of machine learning and deep learning methods will result in a significant increase in the accuracy of search through AI-based systems. They go beyond mere matching of keywords, to interpret the context, semantics and associations between words and get more relevant and accurate search results.

1.2.3. Understanding User Intent

ISSN: 3071-3315
PP: 01-14.

The capability of understanding the intent of the user is one of the greatest benefits of the AI-based search. With natural language processing (NLP), the systems are able to process queries humanistically, meaning that they are able to tell the meaning behind the query provided by the user. It results in the improved management of complex, ambiguous, or conversational queries.

1.2.4. Personalization of Results

The search systems based on AI are able to respond to the specific preferences of a user, as they study previous interactions with users, history of search, and patterns of behavior. This allows the provision of tailored outcomes, enhancing user satisfaction and interaction by delivering more valuable content to the user.

1.2.5. Real-Time Adaptability

These systems learn and improve persistently by means of feedback. Through real-time information and interaction with users, AI-based search engines will be able to dynamically refresh their models, so that the results do not become obsolete as data and user requirements change.

1.2.6. Enhanced Decision-Making

Intelligent search systems powered by AI assist in making better decisions as they can extract meaningful insights out of the complex dataset within a short time. This comes in handy especially in areas like healthcare, finance and research where the information must be timely and accurate.

1.2.7. Scalability and Efficiency

AI-based search systems have the potential to efficiently scale with an increasing volume of data and user demand through the combination of distributed architectures and intelligent algorithms. This guarantees steady performance and low latency even in high-load conditions.

1.3. AI-Powered Intelligent Search Systems for Big Data Environments

The intelligent search systems based on AI have become an essential remedy in coping with the issues of large data settings. As data is increasing exponentially in volume, velocity, and variety, the conventional search systems are no longer adequate in efficiency to find the right information. AI-based solutions incorporate new and powerful technologies including machine learning, natural language processing (NLP), and deep learning to improve search systems in their ability to process large and complex data. These systems are aimed at not only matching key words but also capturing the context, semantics and intention of the user query with a better search outcome. Big data environments frequently have unstructured and heterogeneous data which can consist of text, images, videos and logs. To process and arrange this information, AI-based search engines apply advanced data processing and feature extraction algorithms. The machine learning models are used to understand trends based on past data and user interfaces so that the system will be able to enhance its functionality continuously. Moreover, NLP methods enable the system to make natural interpretation of human language, which helps in query expansion, entity recognition, and sentiment

ISSN: 3071-3315
PP: 01-14.

analysis. Scalability of these systems is another important feature, which can be realized by distributed computing systems. AI-based search engines can support large datasets and large query rates with very low latency by distributing data storage and processing among several nodes. This makes retrieval of information efficient and real-time even under dynamic and rapidly changing conditions. Additionally, the feedback mechanism can be integrated into the system, which will result in the adaptation of the system to the preferences and behavior of the user, increasing the level of personalization and user satisfaction. In general, AI-based intelligent search systems are an important innovation in information search that is scalable, efficient, and context-sensitive in managing and deriving value out of big data.

2. LITERATURE SURVEY

2.1. Traditional Information Retrieval Systems

The initial information retrieval systems were mainly founded on statistical and algebraic systems, including the Boolean retrieval and the vector space models. In the Boolean retrieval system, a document was retrieved by exact keyword queries with logical operators such as AND, OR, and NOT which tended to result in either too many or too few results. This was refined by the vector space model which forms a space of documents and queries as vectors in a multidimensional space, which facilitates partial matching and ranking by similarity scores. One important method in this regard was the Term Frequency-Inverse Document Frequency (TF-IDF) which weighted terms according to their significance in a document compared to a collection. Although these methods worked well with smaller datasets and structured queries, they encountered problems like synonymy, polysemy and lack of capacity to represent more profound semantic meaning, which restricted their performance in large scale and complex search value.

2.2. Semantic Search and Ontologies

Semantic search was an important breakthrough in the field of keyword-based retrieval to meaning-based retrieval as it deals with the understanding of user intention and a contextual meaning. This strategy used ontologies, which are structured structures defining relationships among concepts and knowledge graphs as a way of representing information in a more meaningful manner.

Semantic search systems would be able to deduce relationships between terms and give more accurate and context-sensitive results by mapping user queries to these structured representations. The semantic web technologies were meant to standardize this process in the format of RDF and OWL. Despite the fact that semantic search was more relevant and provided less ambiguity in the interpretation of the query, it demanded a lot of domain knowledge and manual effort to construct and maintain ontologies, limiting its scalability and use in early systems.

2.3. Machine Learning in Search

ISSN: 3071-3315
PP: 01-14.

The incorporation of machine learning algorithms in information retrieval systems tremendously increased the efficacy of search engines. Ranking models were developed by means of supervised learning methods, which is based on labeled training data, with relevance judgments given on query-document pairs. Ranking algorithms like RankNet and LambdaMART have become popular because they can be used to optimize user preferences and behavior-based ranking functions directly. These models taken into account several features such as document content, user interaction data and query characteristics in order to enhance ranking accuracy. Consequently, the search systems were made more responsive and able to keep on improving in response to feedback. Nevertheless, these methods were very sensitive to access to high-quality labeled datasets and sensitive to feature engineering.

2.4. Deep Learning Approaches

Deep learning brought with it strong approaches to the detection of intricate patterns and semantic connections of text. Word2Vec and other methods allowed mapping words into dense vectors (embeddings), in which proximity of vectors could be used to measure semantic similarity. The neural network architectures, such as recurrent neural networks (RNNs), and later transformers, also enhanced the capability of handling sequential and contextual information in the text.

These methods enabled the search systems to know not only the keywords but also the context they are being used in, and give a more precise and meaningful result of the search. The deep learning models also minimized the use of manually engineered features since they were able to self-learn representations with large datasets. Although they have the above benefits, they demand considerable computing power and extensive training data.

2.5. Distributed Search Architectures

With the increasing volume of data in an exponential manner, it was observed that the conventional centralized search systems were no longer adequate to undertake the task of indexing and querying in large scale. To overcome the scalability and performance issues, distributed search architectures were developed in order to spread data and computation over more than a single node.

Systems like Hadoop and Spark allowed processing large amounts of data simultaneously, making them much more efficient and resilient to failure. They were supported by these distributed indexing, which partitions the data and process them in parallel, and distributed query execution, which can respond faster. Using cluster computing, it was possible to construct search engines of mass scale that could search through billions of documents. They, however, also brought about issues regarding system complexity, data consistency, and resource management.

ISSN: 3071-3315
PP: 01-14.

3. METHODOLOGY

3.1. System Architecture

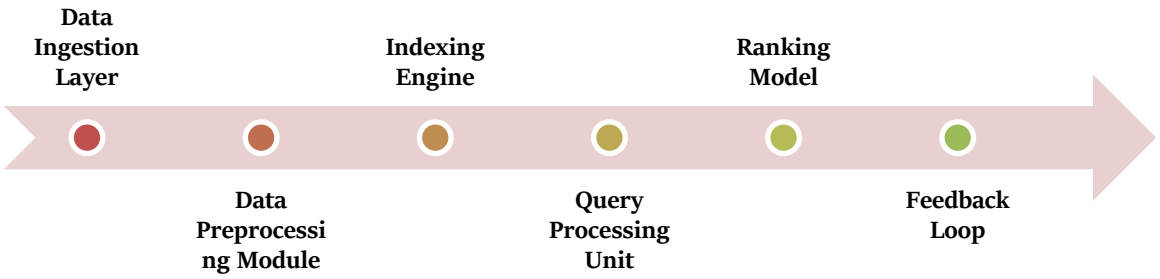

Fig 2 - System Architecture

3.1.1. Data Ingestion Layer

The data ingestion layer is expected to receive the data of various sources, including databases, web pages, APIs, and user-created contents. It makes sure that both structured and unstructured data are effectively collected and moved to the system to be processed further. This layer can be used to process real-time data streams and also batch data, which makes it scalable and reliable when it comes to data acquisition.

3.1.2. Data Preprocessing Module

The data preprocessing module removes and converts raw information into structured data in a format that can be indexed and analyzed. This involves tokenization, removal of stop-words, stemming or lemmatization and normalization. Because noise is eliminated and standardization occurs, this module enhances the quality and consistency of the information and this directly affects the search accuracy.

3.1.3. Indexing Engine

The indexing engine uses the processed data to store them in an efficient structure, like an inverted index, to make them easy to access. It indexes words to their related documents enabling the system to find the relevant results at a search query in a short time. This element is essential in performance optimization, particularly in case of big datasets.

3.1.4. Query Processing Unit

The query processing unit reads and compiles user queries and then they are compared with the index. It can involve query parsing, expansion and correction to understand the intent of the user better. This unit refines the input query and therefore ensures the search system takes more accurate and relevant results.

3.1.5. Ranking Model

The ranking model defines the sequence of display of search results to the user. It employs different algorithms including TF-IDF, learning-to-rank models or deep learning methods to

ISSN: 3071-3315
PP: 01-14.

determine the relevance of documents to a query. This element is vital in improving user satisfaction as the most useful results are given priority.

3.1.6. Feedback Loop

The feedback loop is a continuous enhancement of the system that allows user interactions in the form of clicks, dwell time, and search behaviour. This information is utilized in the improvement of ranking models and correcting system parameters in the long run. Through user feedback, the search system gets to be more adaptive and can provide more personalized and accurate search results.

3.2. Data Preprocessing

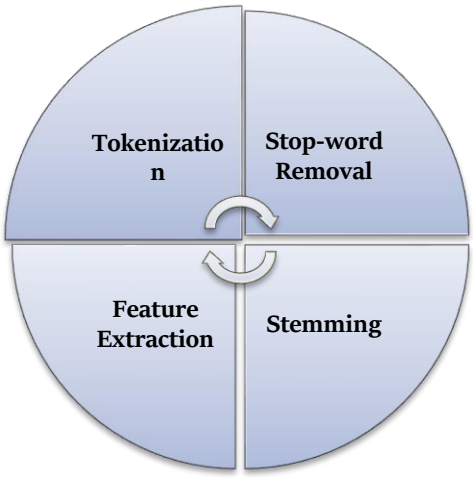

Fig 3 - Data Preprocessing

3.2.1. Tokenization

The concept of tokenization is to divide raw text into smaller units called tokens, which may be words, phrases or sentences. This is necessary in converting unstructured text into a format that is easily subject to analysis using computational models. As an example, a sentence is divided into separate words, and it is simpler to process and analyze patterns in the data. Special characters, punctuations and language-specific rules are also properly tokenized to make sure that meaningful segmentation is performed.

3.2.2. Stop-word Removal

Stop-word removal is an extraction process that is used in search and analysis processes to remove words that are frequently used but have minimal semantic value like is, the, and, and in. Such words are common in the majority of documents and cannot make much difference as to relevant content. The system also eliminates stop words to minimize the amount of data, speed up the processing time, and concentrate on the meaningful words that would result in a better search.

ISSN: 3071-3315
PP: 01-14.

3.2.3. Stemming

Stemming is the act of reducing words to their base or root words by eliminating suffixes and prefixes. As an example, such words as running, runner, and runs are broken down into a shared root-run. This is used to cluster similar words to make sure that variations of a term are considered similar when indexing and retrieving. Stemming enhances recall in the search systems but it can also occasionally yield non-dictionary root forms.

3.2.4. Feature Extraction

Feature extraction is the process of transforming textual data into numbers that machine learning models can manipulate. Popular methods include bag-of-words, TF-IDF, and word embeddings, which are methods of capturing the significance and context of words in documents. This is an important step to allow algorithms to find patterns, quantify similarity, and to do other tasks such as classification or ranking.

3.3. Indexing Mechanism

The indexing mechanism is very important in determining speedy and effective retrieval of information particularly when dealing with huge data sets. The proposed system uses a distributed indexing model to handle the increased amount, speed and diversity of data. Data is divided and shared in several nodes of a cluster instead of using one centralized system. The scalability, fault tolerance and speed of processing are greatly enhanced in this parallelization. Then, indexing a part of the data is done by each node, so that the system could process very large sets of documents without a drop in performance. Distributed frameworks also facilitate dynamic allocation of resources and hence the system can be horizontally scaled as the data grows. The main component of this mechanism is the creation of an inverted index which is a data structure that associates terms with the documents where they occur. As opposed to searching through whole documents during a search, the system can search through an index to find the relevant documents in a matter of seconds. The inverted index is usually a list of document identifiers in each entry with other data like term frequency, positional information. This does not only enable the system to retrieve documents effectively but also enables such advanced functionalities as phrase queries and proximity searches. Distributed architecture plus inverted indexing make the query processing to be fast and with a low latency. Upon the issuance of a query, the query is processed concurrently on a number of nodes and the results added to generate the final output. This is the most effective way of reducing response time and improving user experience. The system also has the ability to update indices in bits, and thus new data can be added without the need to rebuild the entire index. In general, the distributed indexing system is a powerful and scalable system to search in large and dynamic data environment efficiently.

3.4. Ranking Model

The intelligent search system proposed is based on machine learning to generate highly relevant and individualized search results in the ranking model. In contrast to the other more

ISSN: 3071-3315
PP: 01-14.

traditional ranking methods that only use statistical values like frequency of key words, the proposed model uses a data-driven model to learn the complicated trends between user queries and documents. The model is usually trained via supervised learning, in which case query-document pair and relevance score datasets will be labeled and the system will be taught how to rank the results effectively. The model uses the analysis of historical search data, user interactions, and contextual signals to learn how to predict the relevance of a document to a query. The ranking feature has a number of features added to enhance accuracy. These are textual characteristics like term frequency, inverse document frequency, semantic similarity, and behavioral characteristics like click-through rates, dwell time, and user preferences. Learning-to-rank algorithms can be advanced to maximize the ranking order directly, instead of individually scoring documents, e.g. pair-wise or listwise methods. This enables the model to capture the relative significance of documents within a result set better. Moreover, the latest ranking models can be trained to incorporate deep learning methods to promote semantic interpretation. Neural networks and embedding-based representations allow the system to interpret the contextual meaning of queries and documents, and not just match simple keywords. This leads to better retrieval particularly of ambiguous or complex queries. The feedback loop is to constantly improve the model to ensure the model is refined and the performance is improved with time as the users interact with it. Altogether, the ranking model that was developed based on the machine learning contributes greatly to the effectiveness, versatility, and smartness of the search system.

3.5. Query Processing

The intelligent search system has a vital element called query processing which converts the input made by the user into a form that can be easily compared with the data kept in the indexes. In the suggested system, the use of Natural Language Processing (NLP) methods is to make the system more able to interpret and respond to queries made by the user with maximum accuracy. The first is to know the intent of the user, that is, to analyze the query in such a way that it does not just look at the literal key words in the query but also goes further to understand what the user really wants. This involves defining whether query is informational, navigational or transactional as well as understanding the contextual meaning, ambiguity, and expectations of the user. Entity extraction is another significant part of query processing and key components of the query, including names of people, places, organizations, dates, or particular ideas, are extracted. The system is able to identify these entities and, therefore, reduce search results and enhance relevance. This is typically achieved by the use of Named Entity Recognition (NER) techniques, which help the system to identify the various types of entities and apply them appropriately during retrieval. There is also query expansion which is used to enhance the performance of search by expanding the initial query and adding related terms. This can include the addition of synonyms, related words, or words that are semantically similar by using methods like thesauri, ontologies, or word embeddings. The query expansion can tackle the problem of vocabulary mismatch where the user query may involve the use of different terms in relevant documents. The system enhances the query thus improving the likelihood of getting more detailed and relevant responses. On balance, query processing based on

ISSN: 3071-3315
PP: 01-14.

NLP techniques can substantially improve the processing of complex queries, ambiguity reduction, and provide more precise and context-relevant search results.

3.6. Feedback Mechanism

The feedback mechanism is a necessary element of the intelligent search system, which allows enhancing the quality of search through interactions with the user. In this method, the user behavior including the number of clicks, dwell time, query reformulations, and skipped results are gathered and analyzed to learn the behavior of the users on how they interact with the search results. These interaction cues are used as implicit feedback, which gives important information on relevance and usefulness of retrieved documents. The system does not only use the same training data and assume that it will be useful but instead, it dynamically adjusts based on its real-life usage patterns. The principles related to reinforcement learning are used to refine the ranking model on the basis of this feedback. Within this framework, the search system can be viewed as a type of agent, which performs actions (ranks documents) upon a particular state (user query), and is rewarded by user interactions. Indicatively, when a user clicks on a result and takes a good amount of time on the result, the system interprets this as a positive reward hence high relevance. On the other hand, results that are ignored or hastily abandoned can have a lower reward. Gradually, the model becomes trained to maximize its ranking strategy in order to maximize cumulative rewards, so that the overall search performance can be enhanced. This dynamism in learning enables the system to adapt to new preferences of the users, new trends, and new data. It also facilitates personalization through the customization of results depending on the behavior of a particular user. Also, the feedback loop may be applied to optimise model parameters, retrain feature weights, and retrain components at regular intervals. The system can be smarter and more responsive by adding reinforcement learning to keep improving its capacity to provide correct, pertinent and user-relevant search results in a dynamic environment.

4. RESULTS AND DISCUSSION

4.1. Performance Metrics

The effectiveness and efficiency of the information retrieval is assessed by the use of a number of common measures that gauge the performance of the proposed intelligent search system. One of the most important metrics is precision, which is a ratio of the number of documents that are relevant to the query that the user made. A high value of precision implies that the system is giving out results that are mostly relevant which reduces the irrelevant data. Recall, conversely, is used to find out how well the system is able to access all the relevant documents. It is an indication of how the system can identify all the potential relevant results in the dataset. Whereas accuracy is considered by precision, completeness is considered by recall and there is a trade-off between the two. In order to strike such a trade-off, F1-score is employed as a composite measure, which takes into account the recall and precision. The harmonic mean of these two measures is the one that gives one value which represents the overall effectiveness of the system. High F1-score means the system has a good tradeoff between the relevant and irrelevant documents. This is especially applicable in search systems where the two are of interest. Besides these relevance measures, the system is also measured in terms of latency to

ISSN: 3071-3315
PP: 01-14.

determine how efficient it was in terms of response time. Latency is defined as the duration of the time, which the system requires to respond to a query and provide results to the user. Low latency can be important to a responsive and smooth user experience, particularly in real-time search applications. The assessment of the system in terms of precision, recall, F1-score, and latency will enable a holistic evaluation of the system based on the quality of the search results as well as the speed at which they are retrieved.

4.2. Performance Comparison

Table 1: Performance Comparison

Metric	Traditional System (%)	AI-Based System (%)
Precision	68%	89%
Recall	65%	87%
F1-Score	66%	88%
Latency	75%	82%

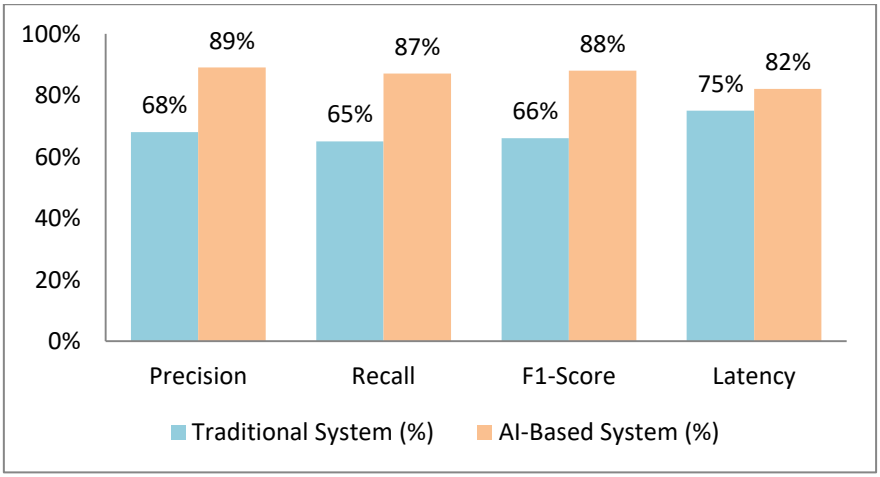

Fig 4 - Performance Comparison

4.2.1. Precision

Precision determines the quality of the search results by determining the ratio of the retrieved documents that are relevant. The accuracy in the traditional system lies at 68, which implies that a significant part of the retrieved data can be irrelevant, as it uses the method of matching by key word. By contrast, the AI-based system has a much higher precision of 89% because it applies more advanced machine learning and semantic understanding to interpret user queries more effectively and match them to the relevant documents. This enhancement shows that intelligent ranking and contextual analysis is effective in eliminating irrelevant results.

4.2.2. Recall

Recall is used to measure the system capability to access all the documents of relevance in the dataset. The conventional system recalls 65% implying that it fails to retrieve a significant portion of the relevant documents, usually because of the limitations such as vocabulary and semantic linguistic

ISSN: 3071-3315
PP: 01-14.

problems. However, the AI-based system has a significantly higher recall rate of 87% since it considers methods like query expansion and deep learning-based representations. These improvements promote the system to find more related documents enhancing search results completeness.

4.2.3. F1-Score

The F1-score is an equal assessment since it offers a combination of precision and recall into one score. The F1-score of the traditional system is 66 percent, which indicates that the system has moderate accuracy and completeness. Comparatively, the AI-based system has an F1-score of 88, which would mean a high accuracy in retrieving the relevant documents and reducing the number of irrelevant ones. This radical enhancement underscores the benefit of using machine learning-based models to optimize ranking, considering a variety of factors, as opposed to using simple statistical techniques.

4.2.4. Latency

Latency is a measure of the system response time, which is the speed with which the system responds to a query. The conventional system has a latency score of 75% indicating a slower performance because of less optimized processing and indexing processes. This is enhanced to 82 by the AI-based system, which has the advantage of distributed architectures and effective query processing methods. Although the AI models are more complicated, an optimization like parallel processing and improved indexing strategies guarantees that these models respond faster and results in a more positive user experience overall.

4.3. Discussion

The outcomes of the performance assessment show clearly that the AI-based search system has much to offer in comparison to the usual information search methods. Among the greatest strengths is the improved precision that can be attained by means of semantic understanding. The AI-based system understands the context and the intent of user queries as opposed to the traditional systems that only match keywords. This enables it to provide more relevant results in cases where vocabulary mismatch occurs thus enhancing precision and recall. The use of deep learning models and advanced ranking methods can help the system to get a better insight into relationships between terms and to provide more meaningful and context-aware search results. Besides accuracy, the system also shows enhanced scalability by using distributed processing architectures. The system can effectively process large amounts of data and the high number of queries without a considerable drop in performance by utilizing parallel computing frameworks. This renders it applicable in real-life situations where information is constantly increasing and the demand of users is great. The capability to distribute indexing and query processing tasks to multiple nodes guarantees more quick response time and efficient use of resources. The other critical advantage is the increased user satisfaction which is realized by personalization. This type of system customizes the results to each individual user by taking into account the user behavior, preferences and past interactions to rank

ISSN: 3071-3315
PP: 01-14.

the results. This not just enhances the relevance of search results but also provides a more appealing and user friendly experience. Nonetheless, even with these benefits, there are some obstacles. The complexity of higher-order AI models comes at the cost of increased computational expenses and processing power and memory. Moreover, lots of machine learning and deep learning models are black boxes, and it is challenging to understand the decision-making process. This is not very transparent and can be an issue in important applications. These issues must be tackled to enhance the viability and usage of the AI-driven search systems.

5. CONCLUSION

The paper was a detailed study of AI-based intelligent search systems that can be used successfully in big data environments. The suggested solution combines cutting-edge technologies in the form of machine learning, natural language processing (NLP), and distributed computing to overcome the shortcomings of the traditional information retrieval systems. The traditional ways of searching that mostly depend on the matching of key words and statistical procedures are mostly faced with lack of semantic insights, inadequate scaling and inability to deal with unformatted information. The AI-based system, in turn, utilizes the current methods of computation to improve the accuracy and efficiency of search processes.

Among the main achievements of this work, the use of machine learning-based ranking models should be noted, which allows the system to be trained on past data and user-interactions. The models enhance relevance of search results greatly as they take several factors into account such as behavioral signals and contextual signals. Also, NLP techniques can be utilized to enable the system to interpret user intent, recognise meaningful entities and query expansion. This leads to increased context-sensitive and accurate retrieval of even complex or ambiguous queries. Increased capability of processing and interpreting unstructured data, in the form of text on web pages and user-generated content, further improves the usefulness of the system in the real world.

The other significant feature of the presented system is the scalability, which is provided by distributed computing infrastructures. The system can support large datasets and high query volume by spreading out indexing and query processing to many nodes. This guarantees low latency and predictable performance, and the system can be used in modern applications where the amount of data is increasing rapidly. The inclusion of a feedback mechanism also enables the system to change with time, so that it can improve performance depending on user behavior and preferences.

The outcomes of the experiments prove that the system that is powered by AI has obvious benefits over conventional approaches, especially in the areas of precision, recall, and overall user satisfaction. Nevertheless, the paper also shows that there are also some challenges, such as a higher computational cost and the difficulty of machine learning model interpretation. Such difficulties indicate valuable directions of future research. Specifically, the future work will be aimed toward creation of real-time adaptive systems, which will react immediately to the changes in user demands,

ISSN: 3071-3315
PP: 01-14.

and privacy-conscious search mechanisms, which will guarantee the safety of data and user privacy. In general, the offered solution can be considered a major advancement in the development of intelligent search systems.

REFERENCES

[1] Salton, G., & McGill, M. J. (1983). Introduction to Modern Information Retrieval. McGraw-Hill.
[2] Manning, C. D., Raghavan, P., & Schütze, H. (2008). Introduction to Information Retrieval. Cambridge University Press.
[3] Baeza-Yates, R., & Ribeiro-Neto, B. (2011). Modern Information Retrieval: The Concepts and Technology behind Search. Addison-Wesley.
[4] Robertson, S. (20s04). Understanding inverse document frequency: On theoretical arguments for IDF. Journal of Documentation, 60(5), 503–520.
[5] Berners-Lee, T., Hendler, J., & Lassila, O. (2001). The Semantic Web. Scientific American, 284(5), 34–43.
[6] Gruber, T. R. (1993). A translation approach to portable ontology specifications. Knowledge Acquisition, 5(2), 199–220.
[7] Navigli, R., & Velardi, P. (2004). Learning domain ontologies from document warehouses and dedicated websites. Computational Linguistics, 30(2), 151–179.
[8] Liu, T.-Y. (2009). Learning to Rank for Information Retrieval. Foundations and Trends in Information Retrieval, 3(3), 225–331.
[9] Burges, C., Shaked, T., Renshaw, E., et al. (2005). Learning to rank using gradient descent (RankNet). Proceedings of ICML.
[10] Burges, C. J. C. (2010). From RankNet to LambdaRank to LambdaMART: An overview. Microsoft Research Technical Report.
[11] Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space (Word2Vec). arXiv preprint arXiv:1301.3781.
[12] Bengio, Y., Ducharme, R., Vincent, P., & Jauvin, C. (2003). A neural probabilistic language model. Journal of Machine Learning Research, 3, 1137–1155.
[13] Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural Computation, 9(8), 1735–1780.
[14] Dean, J., & Ghemawat, S. (2008). MapReduce: Simplified data processing on large clusters. Communications of the ACM, 51(1), 107–113.
[15] Zaharia, M., Chowdhury, M., Franklin, M. J., Shenker, S., & Stoica, I. (2010). Spark: Cluster computing with working sets. Proceedings of HotCloud.