International Journal of Engineering & Scientific Research
Vol. 10 Issue 8, August 2022,
ISSN: 2347-6532 Impact Factor: 6.660
Journal Homepage: http://esrjournal.com, Email: esrjeditor@gmail.com
Double-Blind Peer Reviewed Refereed Open Access International Journal - Included in the International Serial Directories Indexed & Listed at: Ulrich's Periodicals Directory ©, U.S.A., Open J-Gage as well as in Cabell's Directories of Publishing Opportunities, U.S.A

Role of Deep Learning in Information Retrieval for Advanced Database Web Search

Sumathi Rani Manukonda
Master in Computer Science (MS)
University of Bridgeport, Connecticut, USA

Abstract:

Information is the level of abstraction that comes after data and before knowledge. Information retrieval bridges knowledge and information by storing, organizing, representing, maintaining, and disseminating information.The user gives the query to retrieve information from the database.The database community should take the lead in promoting this new trend because it has long concentrated on data-driven applications. Manual information retrieval wastes resources and takes a long time to process, whereas machine learning techniques apply statistical models, which are flexible, adaptable, and quick to train. A refinement of machine learning known as "deep learning" includes hierarchical learning layers, which makes it perfect for demanding tasks. Deep learning has recently skyrocketed in popularity due to its outstanding effectiveness in various complex data-driven applications, including picture classification and speech recognition. Deep learning's success depends on its capacity to consistently acquire distributed representations of natural language expressions like sentences and apply those representations to tasks.

In the past, big feature sets for various information extraction challenges, such as entity mention detection, relation extraction, co-reference resolution, event extraction, and entity linking, have primarily been created by hand. This method suffers from the unseen word/feature problem of natural languages and is constrained by the time-consuming and expensive work necessary for feature engineering for various domains. Deep learning contains a lot of informational resources and big datasets for computation, making it the most excellent option for information retrieval. Firstly, this articlestarts by talking about several uses for deep learning in information retrieval, such as decreasing noise in web searches and gathering accurate results, spotting trends in social media analytics, spotting anomalies in various datasets, and image retrieval, among others. Secondly, this articleintroducesfundamental concepts in deep learning for information retrieval and natural language processing, including word embedding, recurrent neural networks, and convolutional neural networks.Finally, examining database applications that could profit from deep learning techniques and propose potential database-related improvements for deep learning systems.

Keywords:
Machine Learning, Deep Learning, Information Retrieval, Advanced Databases.Natural Language Processing

Introduction:

Information retrieval (IR) is the process of selecting from a pool of information resources those that are most pertinent to a given information demand. The information retrieval system receives a request for information from the user expressed as keywords or a question. The system then accesses the information and knowledge base, discovers relevant results, and then sends the request back to the user. The main issues are expressing

purpose and content, executing intent, and matchingthe content. Document extraction is the simplest method for retrieving information. The matching (relevance) score between the query and each document is determined by computing the cosine similarity between their tf-idf vectors. The term mismatch drawback affects the usual strategy, although it also partially works effectively. In reality, semantic alignment between the query and the page is essential for assisting the user in efficiently finding pertinent information [1]. Semantic matching in web searches has been successfully carried out using deep learning techniques, and a notable increase in relevance has been seen. [2,3]. In IR, there are more challenging tasks, including locating images and responding to inquiries from documents, relational databases, knowledge bases, and papers. The challenge arises from the fact that the representations in the tasks may be unstructured data, structured data, multimedia data, or a combination of these, making direct matching between the jobs' intent and content extremely challenging or impossible.Even though several algorithms for image retrieval, question answering, and other tasks have been proposed, their results were insufficient and tended to be ad hoc. With deep learning serving as the main equipment, considerable advancements in solving the challenges in IR have recently been made.

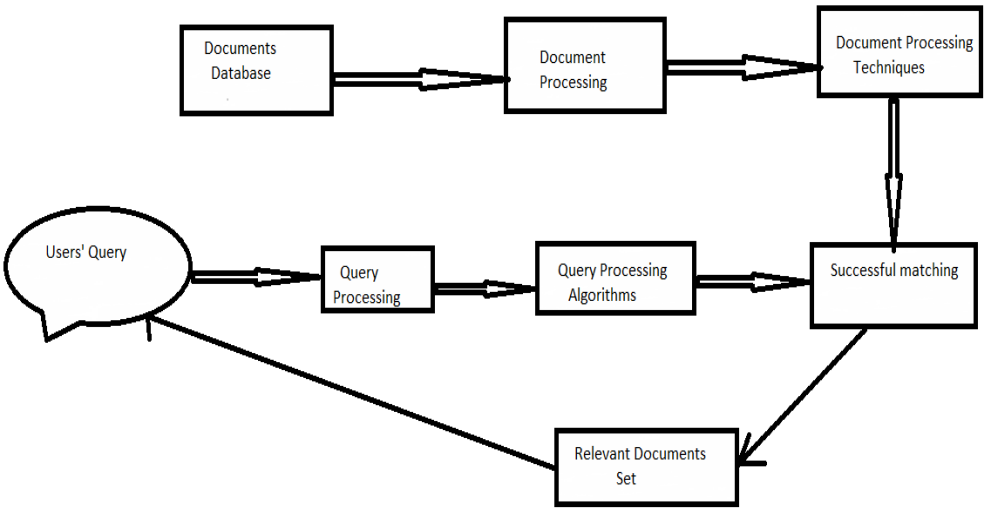

Figure: Information Retrieval Process

We encounter enormous amounts of data daily using social media, websites, and mobile applications; for example, Facebookgenerates 15 GB of data [5]. Unstructured data in online logs, data records, sensor data, etc., constantly expandsand needs to be transformed into valuable information. We gather knowledge from unrestricted data to close the knowledge gap. The user's query is read by the IR system, which searches databases and knowledge bases for information on images, text, sound, and other types of data [6]. This information is then returned to the user. The obtained documents are sorted according to their perceived value for a given query. Indexing, filtering, looking for matches, and ranking the documents are intermediate processes. Indexing, filtering, looking for matches, and ranking the documents are intermediate processes. The documents are indexed using signature files, inverted indices, etc., and the filters eliminate any stop words, white spaces, etc. The query is then searched using brute force, linear search, and other techniques, and the found documents are ordered according to how closely they match the query. The user receives the highest ranked documents in response [5].

In that it is probabilistic (unstructured data) in nature as opposed to deterministic (structured data in tables), it differs from the Database Management System (DBMS). This is so that all relevant documents containing the necessary information are delivered since the IR system scans the documents for the keywords specified by the query. On the other hand, DBMS provides a very particular detailedquery that matches strictly. Its adventure began with library management and has since expanded to include applications for office automation, knowledge extraction, multimedia management, medical information management, etc. Because of the internet, data is growing tremendously daily; thus, it must be handled carefully. Therefore, some kind of automation is necessary to retrieve the data from such a vast collection.

The machine learning and computer vision communities gave rise to deep learning. It has been successfully utilized in various fields, such as NLP. However, deep learning methods haven't been used in much research on conventional database issues. This is mainly because deep learning effectively forecasts uncertain occurrences, but traditional database issues like indexing, transaction, and storage management entail less uncertainty.

Deep learning is effective because it enables the automatic learning of representations of many types of data for various purposes. Real-valued vectors, also known as distributed representations, are the only form in which the learned models can exist. By conducting the matching in IR using vector representations, it is possible to dramatically improve the performance of some IR tasks while also completing others previously thought possible [4]. Additionally, deep learning can answer queries from databases or knowledge bases. Recent research demonstrates that one can train a deep neural network to automatically answer questions from a relational database or knowledge base given only question and answer pairs, the database or knowledge base, and the "grounding" relations between the answers and the database or knowledge base. A semantic parser can be built automatically to analyze the queries with even higher performance. Deep learning has also improved the process of answering common inquiries from texts. This procedure can be described as a matching problem between two demanding syntactic and semantically structured sentences (an inquiry and a response). The technique has the advantage that, in most circumstances, no linguistic competence is required to develop the system [7,8]. The ability to create a system that automatically generates responses to questions in single-turn dialogue using deep learning and a sizable amount of question-answer pair data is another fresh and exciting discovery [9]. This was seen as a highly challenging issue.

Its authenticDeep learning has indeed several completely new prospects for IR. Many brand-new tasks can be taken on, and deep learning can significantly aid in resolving many ongoing issues. Yes, there are undoubtedly many challenges. The biggest hurdle is combining traditional symbolic processing with neural computing (or deep learning), which is crucial for IR. However, databases like knowledge fusion and crowdsourcing have probabilistic problems [10,11]. Numerous fields can benefit from the usage of deep learning technologies.

Related Work:

Finding things with information relevant to a particular query is the process of information retrieval. A set of data can be organized, verified, and stored. Although the information's source (databases) varies, the end user can readily retrieve the desired information. An information source may be structured (relational databases), semi-

structured (XML, scientific data), or unstructured (such as text files, emails, or audio files). Information retrieval tells the end-user how to obtain the desired data, along with information about where it came from and how many sources it had [12].

The most often asked questions in the database fields are the efficiency, speed, and dependability of relational database systems have been covered in several papers, along with object-oriented databases and other database architecture. However, these latter two are still insufficient and could not satisfy the needs of the research community [13]. The best method for optimizing database systems is indexing. In addition, parallelization is one of the best ways to optimize an index. The goal is to accelerate data processing and shorten the time it takes for sophisticated queries to respond. Query optimization is still one of the best ways to do that [14]. A poorly constructed query might lengthen the execution time by increasing the input-output ratio. Afterward, dial back the system. An English-based semantic and syntactic corrector is suggested to address this issue. When a request is delivered to an RDBMS, it is parsed and converted into RDBMS(Relational Database Management System) language, after which the RDBMS creates many execution plans from which the RDBMS optimizer selects and executes the most appropriate one [15].

Structured data contained in databases is accessed through natural language interfaces (NLI). There are many other NLI types, but the one that receives the most significant is natural language interfaces to databases (NLIDB), which uses a relational database to store structured data. The drawback of this approach is that most non-computer scientists find SQL too challenging. In that case, we require an“RDBMS comprehensible” representation and "human-understandable.” Natural language interfaces to knowledge bases (NLIKB), which use an ontology to organize information, are another NLI. Natural language Interfaces can interface with all RDBMS and employ close algorithms to interpret NLI queries and map them to RDBMS queries. Utilizing local parallel index partitioning, queries in database systems are optimized [16,17].According to the types of connections between layers, deep learning models can be divided into three categories: feedforward models (direct link), energy models (undirected connection), and recurrent neural networks (recurrent connection). Convolution Neural networks (CNN) and other feedforward models transport input data across each layer to retrieve high-level information [18]. The most advanced model for many computer vision applications is CNN. Deep Belief Network (DBN) energy models are frequently used to pre-train feedforward and other models, such as energy models. Sequential data modeling frequently uses recurrent neural networks (RNNs). Popular RNN applications include language modeling and machine translation [19].

The model parameters used in the transformation layers must be taught before a deep learning model can be used. Finding parameter values that minimize the difference (loss function) between the expected output and the actual output is the training's primary purpose.The training algorithm that receives the most usage is stochastic gradient descent (SGD). SGD computes the gradients concerning the loss function, initializes the parameters with random values, and then iteratively refines them. According to the three model categories mentioned above, there are three popular gradient calculation algorithms: Back Propagation (BP), Contrastive Divergence (CD), and Back Propagation Through Time (BPTT). These methods can be assessed by traversing the graph in particular ways by thinking of the layers of a neural network as nodes.

Machine Learning in Information Retrieval and Databases:
Machine learning algorithms improve IR's effectiveness and efficiency and demonstrate the entire IR process. Unsupervised learning and supervised learning are the two categories into which machine learning algorithms fall. In Supervised Learning, various methods like Multiple Regression Analysis, k-Nearest Neighbor (KNN), Naive Bayes, Random Forest, Neural network and deep learning, and Support vector machine are used in information retrieval for different functions. In Unsupervised Learning, other methods can be used in Machine Learning like k-means Clustering, Hierarchical Clustering, Principal Component Analysis, and Independent Component Analysis.

The user's simple query no longer yields the desired results. Therefore, the query needs to be changed to include related phrases to get better results. By learning from falsely labelled documents and utilizing decision trees to produce Boolean queries, the authors have presented a method for expanding Boolean searches. The ranking is based on query quality predictors [20]. Based on locally taught word embedding, a query expansion technique was created that accurately captured the subtleties of domain-specific language. Instead of using topically constrained corpora, they recommended using sizable topically unrestrained corpora [21]. To improve the answer, support vector machines have been employed to deliver pertinent feedback for the queries for which we have fewer retrieved documents [22].

Medical diagnostic systems are now using machine learning techniques to prioritize requests while making decisions in an emergency. To find similar cases in the databases, the authors employed conceptual IR, or textual processing of clinical data such as diagnosis, principal complaint, etc. [23]. Using the top k cases and a voting system, the request's priority and location for the patient were automatically determined. According to the researchers, using decision trees with random forestsproduced accurate categorization findings. Additionally, they discovered that the semantic approach is quicker and more effective than text mining alternatives. The study created two re-ranking techniques for medical decision support systems [24]. Principal component analysis was suggested for content-based picture retrieval to decrease the dimensionality of the search by utilizing a set of prototype photos [25]. The match is discovered by creating the query image's projection vector and comparing it to the database images.

For independent text classification, the Naive Bayes approach for text retrieval has been suggested [26]. The document images were subjected to learning approaches by the authors for categorization and retrieval. The photos have various patch code phrases that were utilized to retrieve them. A histogram of patches from each partition is constructed after the annotated images are recursively divided into horizontal and vertical divisions. When trained with a random forest classifier, this histogram produced perfect precision and recall. To find related semantic images to the same tree node, image retrieval was carried out using the random forest machine learning technique, which divided the tree nodes using visual features and image labels to supervise the division. The querying image's semantic neighbour set is located initially, and the ranking is carried out with a semantic similarity measurement between the query image and the images in the semantic neighbour set [28].

The old search engines employed ranking algorithms that improperly classified the web pages and did not, thus, provide the appropriate web pages [29]. The authors

consequently proposed an intelligent cluster search engine employing k-means clustering. Comparing co-occurring terms and document grouping helped the IR approach. The study applied machine learning technology for information searching [30]. The suggested learning method has created a navigational map of the World Wide Web. The web pages were mapped using a self-organizing map, and feature maps were used to identify their relationship to these keywords. The structure for aiding users with IR was then developed using these maps. Combining translation resources to create cross-lingual IR has been done using the learning-to-rank method. While the query is made in a different language, cross-lingual IR searches for information in one language. Employing translation data from various translation resources, monolingual IR features have been used to map to the cross-lingual IR. It has been demonstrated that mastering ranking techniques have enhanced CLIR performance [31].

The hierarchy for browsing was automatically formed by clustering these documents based on the retrieved attributes [32]. The authors built the software libraries by first aggregating the characteristics from the natural language documents using the indexing technique. Lexical data has been examined using a probabilistic approach for software clustering. Six different vocabularies—class, method, comments, and source code statements—were used to obtain information, and an algorithmic weighting technique was used to determine how much each vocabulary contributed [33]. Utilizing an iterative expectation-maximization approach, these weights are tuned. The similarity between classes has been computed using the vector space model to create software clusters.

To detect anomalies in musical datasets, unsupervised learning was used. Electrical engineering, psychology, musicology, computer science, and engineering are some of the combined subjects to make up music IR [34]. They used a statistical model to obtain the music data, and common traits were then extracted. Without training data, the clips are correctly categorized as corrupted, distorted, or mislabelled in the dataset of anomalous music genres. Based on segmentation issues and missing and incorrect meta-data, the broadcast and CD collections have been cleaned using the probability density function [35]. By training, they discovered the relationship between music characteristics and meta-data and the unlikely music features by training conditional and unconditional densities.

Machine learning (ML) techniques to improve databases haverecently received much attention. Conventional database optimization methods like cost estimation, join order selection, knob tuning, index, view advisor, and others are based on empirical approaches and specifications and need human intervention (such as DBAs) to maintain and tune the databases [36]. As a result, existing practical methodologies, particularly those used in the cloud, cannot provide the high performance needed for large-scale database instances, a variety of applications, and a wide range of users. Fortunately, there are learning-based strategies that can solve this issue. For example, deep reinforcement learning can be used to tune database knobs, reinforcement learning can be used to optimize join order selection, and deep reinforcement learning can increase cost estimation accuracy [37,38].

Regression problems are a standard paradigm for database issues. A regression model (such as a deep learning model) can be used for cardinality estimation, which seeks to estimate the cardinality of a query. A regression model can be used to estimate the benefit of producing an index (or a view), which is the goal of index/view benefit estimation [39,40]. A regression model can be used to estimate performance based on

query and concurrency features, and latency prediction aims to assess the execution time of processing a query. The database must also be actively optimized by anticipating incoming queries. To optimize query performance or resource utilization, these prediction challenges pinpoint the temporal workload patterns [41Manyany machine learning techniques address these prediction issues, including reinforcement learning for workload scheduling and cluster-based algorithm for trend prediction [42].

Deep Learning in Information Retrieval:

The amount of data is increasing dramatically every day, and learning from such a vast database is a complex volume (scalable data), velocity (data expansion), variety (different sources), and veracity; typical machine learning algorithms are not appropriate for extensive data (uncertain data). Deep learning techniques were developed [43]. As a result, it eliminates the effects of gradients and is hence more suited for usage with unprocessed, highly dimensional data. It learns from low-level to high-level features; for example, it can learn raw pixel input as colour information. In the next layer, it may go up to the edge of the object using the data from the previous layer. Deep learning has its roots in neural networks, which combine feed-forward NN with many hidden layers. Deep learning automatically selects high-dimensional, heterogeneous, raw data without manual selection. As a result of data-driven and computationally intensive activities, machine learning has moved in the direction of deep learning.

The authors created a model for sentence embedding using RNN with LSTM cells that extracted the information from each word of the sentence and embedded it into a semantic vector due to the general constraint of available human-labelled data for an extensive database. The model was trained using a poorly supervised approach, whichaccumulated user input until the last word of the phrase using its long-term memory [44]. The hidden layer automatically eliminated unnecessary words while maintaining the crucial ones, providing the semantic representation of the entire text. It demonstrated how the semantic vector changed over time and how it only incorporated vital information from any new input. Additionally, the discovered words automatically triggered the RNN-LSTM cells that dealt with related topics. The LSTM-RNN for document retrieval was helped by automatic topic allocation and word detection. The proposed approach performed better than the paragraph vector approach. Deep learning vectors have been constructed to train high-quality vector representations for many unstructured terms. The query expansion strategy also receives a significant amount of term relationships. The technique was empirically examined, demonstrating that it produced superior outcomes to alternative expansion models [45].

Different deep learning techniques used in the IR are DNN (Deep Neural Networks), RNN (Recurrent Neural Networks), CNN (Convolution Neural Networks), and Autoencoder. DNN consists of several hidden layers, eachhundredonline processing elements. It uses neurons from several layers and many input features to automatically extract information from different hierarchical levels. RNN accepts sequential data as input and, by enabling connections between neurons in the same hidden layers, creates a directed cycle. Adopting long short-term memory (LSTM) might mitigate the vanishing gradient problem. Several convolutional layers make up CNN, including subsampling layers to condense the size of the feature map and a collection of filters with narrow receptive fields and learnable parameters. The feature maps are combined to create completely connected layers to create the output.

Deep NN has been proposed for software bug localization to find papers containing bugs [46] automatically. The lexical mismatch is the main obstacle to localizing software bugs. They employed DNN to learn the lexical mismatch in bug reports and source files and a vector space model for textual similarity.

To automatically label the web image data, the authors created a weakly supervised deep learning algorithm [47]. To improve labeling accuracy at the group level and attention, they applied two strategies: random grouping and piling various images into one training instance. It outperformed the jittery signals from the mislabelled photos. The authors created an end-to-end text reading pipeline to detect and recognize text in photographs of natural scenes. CNN was employed for recognition, and a mechanism for region-based detection was applied. Without human labeling, the networks were entirely created by a synthetic text creation engine [48]. Comparing the suggested system to alternative approaches for all standard datasets, it performed well for text and picture retrieval.

When Deep learning outperformed machine learning techniques when it came to creating chemically sound and synthetically feasible compounds with the proper properties for drug discovery, work made a method to use auto-encoders to automate molecular design [50]. The network was fed hundreds of different structures to create a collection of related functions. The DNN successfully created new compounds with drug-like qualities by employing vector decoding, perturbing well-known chemical structures, and interpolating between chemical structures [49].

Deep Learning in NLP (Natural Language Processing):

The fundamental problems can be found in any computational linguistic system. Basic knowledge of the underlying language is necessary to accomplish any linguistic task, including translation, text summary, image captioning, and others. Language modeling, morphology, parsing, and semantics are the four fundamental divisions of this concept. Language modeling can be seen from two perspectives. It first establishes which words come after which. By extension, however, this can be seen as figuring out what words mean because each word has only a limited amount of meaning and only fully contributes to a sentence when interacting with other words. The study of word formation is known as morphology. To depict tense, gender, plurality, and other linguistic constructions, it considers the roots of words and uses prefixes, suffixes, compounds, and other intra-word devices. When parsing, the terms that change other words and create constituents are considered, creating a sentential structure.

Language modeling is possibly the most significant NLP task. A crucial component of practically every NLP application is language modeling (LM) [82]. Language modeling is making a model anticipate words or essential linguistic elements based on existing words or elements. Predictive text entry is helpful for applications where users type input because it enables quick text entering. However, its strength and adaptability come from the fact that it can implicitly understand the syntactic and semantic connections between words or parts in a linear neighbourhood, making it helpful for tasks like text summarization or machine translation. These computers can produce more applicable, human-sounding statements by using prediction.

Concerns were raised about statistical language models' incapacity to deal with synonyms or out-of-vocabulary (OOV) phrases that weren't present in the training corpus.

The improvement of the problem-solving procedure was facilitated by developing the neural language model [83]. The LM community adopted ANNs immediately and continued to create complicated models, many of which were summarized by the author [84]. However, it took much of NLP another decade to use extensively.

A CNN in LM recently replaced the pooling layers with fully-connected one pooling layers were recently replaced with fully-connected layers by a CNN in LM[85]. Like the pooling layers, these layers enabled the feature maps to be shrunk into lower dimensional spaces. However, fully-connected layers preserve some references to the location of such features, whereas any connections in pooling layers are lost. Three distinct architectures were used: a multilayer perceptron CNN (MLPConv) with small MLP filters rather than just linear ones; a multilayer CNN (ML-CNN) with multiple convolutional layers stacked on top of one another; and a combination of these networks called COM with variable kernel filter sizes (if there were three and five) [86].

The findings demonstrated that stacking convolutional layers was detrimental to LM but decreased confusion using MLPConv and COM. Even better outcomes were obtained when MLPConv was used with the different COM kernel sizes. Analysis revealed that the networks had picked up some word patterns, like "as... as," on their own. Finally, the study demonstrated that long-term dependencies in phrases might be detected using CNNs. The closest words were the most significant, but the terms farther away were also important.

The study of what words imply falls under the category of semantics. It considers the specific word meanings, how they relate to and alter other words, the context in which they appear, and a certain amount of general knowledge, or "common sense." Each of these locations has a sizable amount of overlap with the others. As a result, a number of the analysed models can be grouped into several portions. Therefore, they are discussed in the most relevant chapters, where they act.

Even while studying the **fundamentals of NLP**and comprehending how brain models operate, engineering, which prioritizes practical applications over purely philosophical and scientific investigation, sees it as worthless in and of itself. Here is a summary of current methods for several instantly practical NLP tasks. It should be noted that only problems with text processing are covered here; verbal speech processing is not addressed. Speech processing is often regarded as a separate science with many similarities to the study of NLP because it involves knowledge of various other subjects, including auditory processing [65,66].

Information retrieval (IR) systems are designed to assist users in locating the appropriate information in the most practical format at the right moment. The rating of documents about a query string in terms of relevance scores for ad-hoc retrieval tasks, similar to what occurs in a search engine, is one of several IR difficulties that must be addressed [59]. To calculate relevance scores, deep learning models for ad-hoc retrieval compare the texts of queries with document texts. As a result, these models must concentrate on creating representations of the interactions between specific terms in the query and the documents. While some interaction-focused approaches build local interactions directly and then use deep neural networks to learn how the two pieces of text match based on word interactions, some representation-focused methods build deep learning models to produce good representations for the texts and then match the

representations directly [67,68]. Finding how each word in the query links to different parts of the content is helpful when trying to match a lengthy document to a brief query because the relevant piece may appear anywhere in the long document and be distributed [69].

Most contemporary neural IR models are re-rankers for documents that a first-stage effective traditional ranker has determined to be relevant to a query rather than end-to-end relevance rankers. It is impossible to use such ANNs to rank a whole collection of documents since the representations that the neural re-rankers learn are dense for both documents and questions, meaning that most documents in a collection appear relevant to a query.In contrast, the author introduced SNRM PRF, a separate neural ranking model that trained sparse representations for both queries and documents, replicating what conventional methods accomplish [70]. It makes sense for query representations to be denser because queries are substantially shorter than documents and carry far less information than documents do. This was accomplished by combining a sparsity target with hinge loss during training. For questions and documents, in particular, an n-gram encoding was employed. Each word's embedding was run through a separate MLP (multilayer perceptron) before average pooling was applied on top.

To improve three already competitive neural ranking architectures for ad-hoc document ranking, the author retrieved query term representations from two pre-trained contextualized language models, ELMo(Embedding from Language Models) and BERT (Bidirectional Encoder Representations from Transformers) [71,72,73]. One of these designs was DRMM [74]. They also provided a joint model that combined these structures with BERT's classification vector to maximize the advantages of both strategies. The authors' [71] CEDR (Contextualized Embedding for Document Ranking) approach enhanced all three previous models' performance, using BERT's token representations to provide cutting-edge results.

Text classification, or the categorization of free-text texts into predetermined classifications, is another traditional application of NLP. There are several uses for document classification. The author was the first to implement pre-trained word vectors in a CNN for classification at the sentence level [76]. The author's work [76] was inspiring and demonstrated how straightforward CNNs, consisting of a single convolutional layer followed by a dense layer with dropout and softmax output, could perform superbly on various benchmarks with minimal hyperparameter modification. On 4 out of 7 different tasks posed as sentence classification, including sentiment analysis and question categorization, the CNN models suggested were able to surpass state of the art; the author demonstrated that networks with plenty of convolutional layers effectively classify documents. This approach was initially trained without the softmax regression output layer because it was unrelated to the labeled or classification element of the problem. Once both halves of the architecture had undergone pre-trained, they were joined and trained using backpropagation and quasi-Newton techniques, just like a typical deep neural network. On four document datasets, the article used BERT to produce cutting. The author used a hybrid architecture integrating a deep belief network and regression [78, 79]. In a deep belief network, pairs of hidden layers are modeled after limited Boltzmann machines, which are trained via unsupervised learning and are intended to increase or decrease the dimensionality of the input. This was accomplished by repeatedly employing forward and backward propagation on the data until a minimum energy-based loss was discovered. This was accomplished by repeatedbackpropagationof the data until a minimum energy-based

loss was found. This approach was initially trained without the softmax regression output layer because it was unrelated to the labeled or classification element of the problem [80]. Once both halves of the architecture had undergone pre-trained, they were joined and trained using backpropagation and quasi-Newton techniques, just like a typical deep neural network [81].

Although deep learning holds promise for various NLP applications, including text classification, it is not yet without challenges. According to research [75], gradient boosting trees are superior to neural networksCNNs and LSTMs for classifying lengthy full-length books by genre.

Deep Learning in Databases:

Database applications may not look like deepapplications likecomputer vision and NLP. The fundamental principle of deep learning, referred to as feature (or representation) learning, can be used for various purposes [51]. Intuitively, we may compute entity similarity, do clustering, train prediction models, retrieve data with multiple modalities, etc., once we have effective representations for entities, such as images, words, table rows, or columns.

Since natural language **query interfaces** are highly desirable, especially for non-technical database users, they have been tried for decades. The semantics of natural language queries are complex for database systems to parse (or understand) [52]. Deep learning models recently attained cutting-edge performance for NLP workloads. Additionally, it has been demonstrated that RNNs can learn structured output. One alternative is to build SQL queries using RNN models for processing natural language queries and then enhance them using current database techniques. The difficulty is that the model must be trained on a substantial number of (labeled) training examples. Using a modest dataset to train a baseline model and subsequently improving it based on user feedback is one potential method [53] generated SQL query, amend the query generated. This input effectively acts as labeled data for later training.

Query plan optimization is a common issue with databases. The query plan is generated by most modern database systems using intricate heuristics and cost models. Each query plan of a parametric SQL query template, according to the author [55], has an optimality area. The ideal query plan stays the same as long as the SQL query's parameters fall within this range. To put it another way, query plans are unaffected by minor changes in the input parameters. As a result, we can train a query planner that creates (similar) plans for fresh (similar) questions by learning from a set of pairs of SQL queries and optimal strategies.To further explain, we may train an RNN model that takes the components of a SQL query and meta-data (such as the primary key and buffer size) as input and outputs a tree structure that represents the query plan [54]. The model might be trained online using reinforcement learning (similar to AlphaGo), with the execution time and memory footprint serving as the incentive [56]. Be aware that methods that rely on deep learning models might not be particularly successful. First, a probability-based query plan is developed, likely to contain grammatical errors. Additionally, not all query patterns may be covered by the training dataset; for example, some predicates may be absent. A better approach to solving these issues would be to combine database instances and deep learning, for example, by employing some heuristics to detect and rectify language faults.

In database systems, **spatial and temporal data** are popular data types frequently utilized in predictive analytics, progression modeling, and trend analysis [62]. Moving

objects are often mapped into rectangular blocks while processing spatial data. If each block is thought of as a pixel in a single image, deep learning methods, such as CNN, could be used to extract the spatial proximity between adjacent blocks. To predict traffic congestion for a future time point, for example, we could develop a CNN model to capture the density correlations of neighboring locations if we have real-time position data (such as GPS data) of moving items [63]. Deep learning models, like RNN, can be created to model time dependence and predict the occurrence in a future time point when temporal data is modeled as features over a time matrix. An illustration of this would be modeling illness progression based on past medical information, where clinicians would seek to predict the emergence of a specific severity of a known condition [64]. Since most healthcare data is time-series, deep learning can significantly advance the interpretation of healthcare data. Many spatial-temporal problems, such as traffic flow prediction [57], journey duration estimate [58,60], driver behavior analysis, geospatial aggregation querying, etc., have been addressed using deep learning models, including CNN and RNN. A thorough analysis of current developments in deep learning for spatial-temporal data is offered [61].

Conclusion:

This article explores how various machine learning techniques might improve the IR process. Different learning algorithms help classify and rate the proper papers in the most appropriate ways. Deep learning and databases are covered in this article. It is challenging to extract the information users require from the vast database due to the growing amount of data. While deep learning helps learn efficient representation for data-driven applications, databases have a variety of strategies for improving system efficiency. These two disparate sectors share several methods for enhancing the system's performance. This article also discusses potential database-based deep learning system enhancements that might be made and research issues related to the use of deep learning in database applications. We anticipate a seamless integration of machine learning or deep learning with database technology to provide database systems more autonomy and capacity to learn, optimize, and support complex analytics and predictions beyond data aggregation. Additionally, deep learning methodologies can address IR issues. Significant issues with deep learning for information retrieval are discussed in this article, along with opportunities.

References:

[1] H. Li and J. Xu. Semantic matching in the search. Foundations and Trends in Information Retrieval, 7(5):343–469, 2014.
[2] P.-S. Huang, X. He, and J. Gao. Learning deep structured semantic models for web search using clickthrough data. In Proceedings of CIKM'13, 2013.
[3] A. Severyn and A. Moschitti. Learning to rank short text pairs with convolutional deep neural networks. In Proceedings of SIGIR'15, 2015.
[4] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. Nature, 521(7553):436–444, 2015.
[5] Kanimozhi, S. and Padmini Devi, B., "A Novel Approach for Deep Learning Techniques Using Information Retrieval from Big Data" International Journal of Pure and Applied Mathematics, 118(8), 2018, pp. 601-606.
[6] Guan, S. and X. Zhang. "Networked Memex Based on Personal Digital Library." Encyclopedia of Networked and Virtual Organizations. IGI Global, 2008, pp. 1044- 1051.
[7] B. Hu, Z. Lu, H. Li, and Q. Chen. Convolutional neural network architectures for matching natural language sentences. In Proceedings of NIPS'14, 2014.

[8] M. Wang, Z. Lu, H. Li, and Q. Liu. Syntax-based deep matching of short texts. In Proceedings of IJCAI'15, 2015.
[9] O. Vinyals and Q. V. Le. A neural conversational model. arXiv:1506.05869, 2015.
[10] X. L. Dong, E. Gabrilovich, G. Heitz, W. Horn, K. Murphy, S. Sun, and W. Zhang. From data fusion to knowledge fusion. PVLDB, 7(10):881–892, 2014.
[11] B. C. Ooi, K. Tan, Q. T. Tran, J. W. L. Yip, G. Chen, Z. J. Ling, T. Nguyen, A. K. H. Tung, and M. Zhang. Contextual crowd intelligence. SIGKDD Explorations, 16(1):39–46, 2014.
[12] Lal, N., Qamar, S. and Shiwani, S. Information Retrieval System and Challenges with Dataspace. International Journal of Computer Applications, 147, 23-28, 2016.
[13] Chakraoui, M. and El Kalay, A. Efficiency of Indexing Database Systems and Optimising Its Implementation in NAND Flash Memory. International Journal of Systems, Control, and Communications, 7, 221-239, 2016.
[14] Chakraoui, M. and El Kalay, A. Optimization of Local Parallel Index (LPI) in Parallel/Distributed Database Systems. International Journal, 11, 2755-2762, 2016.
[15] Chakraoui, M., El Kalay, A. and Mouhni, N. Tuning Different Types of Complex Queries Using the Appropriate Indexes in Parallel/Distributed Database Systems. International Journal, 11, 2267-2274, 2016.
[16] Zhu, Y.J., Yan, E. and Song, I.-Y. A Natural Language Interface to a Graph-Based Bibliographic Information Retrieval System. Data & Knowledge Engineering, 111, 73, 2017.
[17] Li, F. and Jagadish, H.V. Constructing an Interactive Natural Language Interface for Relational Databases. Proceedings of the VLDB Endowment 8, 73-84, 2014.
[18] S. Cai, Y. Shu, W. Wang, and B. C. Ooi. Isbnet: Instance-aware selective branching network. CoRR, abs/1905.04849, 2019.
[19] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. Nature, 521(7553):436–444, 2015.
[20] Kim, Y., Seo, J., Croft, W.B., "Automatic Boolean Query Suggestion for Professional Search." In: Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 825–834, 2011.
[21] Diaz, F., Mitra, B., Craswell, N.: "Query expansion with locally-trained word embeddings." arXiv preprint arXiv:1605.07891, 2016.
[22] Drucker, H., Shahrary, B. and Gibbon, D.C., "Support Vector Machines: Relevance Feedback and Information Retrieval," Info,"ation Processing & Management, 38(3), pp.305- 323, 2002.
[23] Pollettini, J.T., Pessotti, H.C., Filho, A.P., Ruiz, E.E.S., Junior, M.S.A., "Applying Natural Language Processing, Information Retrieval and Machine Learning to Decision Support in Medical Coordination in an Emergency Medicine Context," IEEE,"8th International Symposium on Computer-Based Medical Systems, pp. 316-319, 2015.
[24] Song, Y., He, Y., Hu, Q. and He, L., "ECNU At 2015 CDS Track: Two Re-Ranking Methods in Medical Information Retrieval." In Proceedings of the 2015 Text Retrieval Conference, 2015.
[25] Sinha, U., and Kangarloo, H., "Principal Component Analysis for Content-Based Image Retrieval,"Radi," graphics, 22(5), pp.1271-1289, 2002.
[26] Lewis, D.D. "Naive (Bayes) at Forty: The Independence Assumption in Information Retrieval," In E," open conference on machine learning, Springer, Berlin, Heidelberg, pp. 4-15, 1998.
[27] Kumar, J., Ye, P. and Doermann, D., "Learning Document Structure for Retrieval and Classification," In Proceedings of the 21st International Conference on Pattern Recognition, pp. 1558-1561, 2012.

[28] Fu, H. and Qiu, G., "Fast Semantic Image Retrieval Based on Random Forest," In Proceedings of the 20th ACM International Conference on Multimedia, pp. 909-912, 2012.
[29] Sathya, M., Jayanthi, J. and Basker, N., "Link-based K-Means Clustering Algorithm for Information Retrieval." In International Conference on Recent Trends in Information Technology, pp. 1111-1115, 2011.
[30] Yang, H.C., Lee, C.H., "Mining Unstructured Web Pages to Enhance Web Information Retrieval," International Conference on Innovative Computing, Information and Control, IEEE, 1, pp. 429-432, 2006.
[31] Azarbonyad, H., Shakery, A. and Faili, H., "A Learning to Rank Approach for CrossLanguage Information Retrieval Exploiting Multiple Translation Resources." Natural Language Engineering, pp.1-22, 2019.
[32] Maarek, Y.S., Berry, D.M. and Kaiser, G.E., "An Information Retrieval Approach for Automatically Constructing Software Libraries." IEEE Transactions on Software Engineering, 17(8), pp.800-813, 1991.
[33] Corazza A., Di Martino S., Maggio V., Scanniello G., "Combining Machine Learning and Information Retrieval Techniques for Software Clustering." In: Moschitti A., Scandariato R. (eds) Eternal Systems. Eternals, Communications in Computer and Information Science, 255, Springer, Berlin, Heidelberg, pp. 42-60, 2011.
[34] Lu, Y.C., Wu, C.W., Lu, C.T. and Lerch, A., "An unsupervised approach to anomaly detection in music datasets." In Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval, pp. 749-752, 2016.
[35] Hansen, L.K., Lehn-Schiøler, T., Petersen, K.B., Arenas-Garcia, J., Larsen, J. and Jensen, S.H., "Learning and Clean-up in a Large Scale Music Database," In 2007 15th European Signal Processing Conference, pp. 946-950, 2007.
[36] D. V. Aken, D. Yang, S. Brillard, A. Fiorino, B. Zhang, C. Billian, and A. Pavlo. An inquiry into machine learning-based automatic configuration tuning services on real-world database management systems. VLDB, 14(7):1241–1253, 2021.
[37] A. Dutt, C. Wang, A. Nazi, et al. Selectivity estimation for range predicates using lightweight models. VLDB, 12(9):1044–1057, 2019.
[38] R. Marcus and O. Papaemmanouil. Deep reinforcement learning for join order enumeration. In SIGMOD 2018, pages 3:1–3:4, 2018.
[39] J. Sun and G. Li. An end-to-end learning-based cost estimator. PVLDB, 13(3):307–319, 2019.
[40] J. Sun, G. Li, and N. Tang. Learned cardinality estimation for similarity queries. In SIGMOD, pages 1745–1757, 2021.
[41] L. Ma, D. V. Aken, A. Hefny, G. Mezerhane, A. Pavlo, and G. J. Gordon. Query-based workload forecasting for self-driving database management systems. In SIGMOD, pages 631–645, 2018.
[42] C. Zhang, R. Marcus, A. Kleiman, and O. Papaemmanouil. Buffer pool aware query scheduling via deep reinforcement learning. CoRR, abs/2007.10568, 2020.
[43] Hinton, G.E., Osindero, S. and Teh, Y.W., "A Fast Learning Algorithm for Deep Belief Nets," Neural computation," 18(7), pp.1527-1554, 2006.
[44] Palangi, H., Deng, L., Shen, Y., Gao, J., He, X., Chen, J., Song, X., and Ward, R., "Deep Sentence Embedding Using Long Short-Term Memory Networks: Analysis and Application to Information Retrieval," IEEE/ACM Transactions on Audio, Speech and Language Processing, 24(4), pp.694-707, 2016.
[45] Almasri, M., Berrut, C., Chevallet, J.P. "A Comparison of Deep Learning Based Query Expansion with Pseudo relevance Feedback and Mutual Information." In: European Conference on Information Retrieval, pp. 709–715, 2016.

[46] Lam, A.N., Nguyen, A.T., Nguyen, H.A. and Nguyen, T.N., "Combining Deep Learning with Information Retrieval to Localize Buggy Files for Bug Reports (N)." In 2015 30th IEEE/ACM International Conference on Automated Software Engineering (ASE), pp. 476-481, 2015.
[47] Zhuang, B., Liu, L., Li, Y., Shen, C. and Reid, I., "Attend In Groups: A Weakly-Supervised Deep Learning Framework for Learning from Web Data." In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1878-1887, 2017.
[48] Jaderberg, M., Simonyan, K., Vedaldi, A. and Zisserman, A., "Reading Text in the Wild with Convolutional Neural Networks," International Journal of Computer Vision, 116(1), pp.1-20, 2016.
[49] Gómez-Bombarelli, R., Wei, J.N., Duvenaud, D., Hernández-Lobato, J.M., SánchezLengeling, B., Sheberla, D., Aguilera-Iparraguirre, J., Hirzel, T.D., Adams, R.P. and AspuruGuzik, A., "Automatic Chemical Design Using a Data-Driven Continuous Representation Of Molecules", ACS central science, 4(2), pp.268-276, 2018.
[50] Blaschke, T., Olivecrona, M., Engkvist, O., Bajorath, J. and Chen, H., "Application of Generative Autoencoder in De Novo Molecular Design. Molecular informatics, 37(1-2), p.1700123, 2018.
[51] W. Wang, X. Yang, B. C. Ooi, D. Zhang, and Y. Zhuang. Effective deep learning-based multi-modal retrieval. The VLDB Journal, pages 1–23, 2015.
[52] F. Li and H. Jagadish. Constructing an interactive natural language interface for relational databases. PVLDB, 8(1):73–84, 2014.
[53] I. Sutskever, O. Vinyals, and Q. V. Le. Sequence to sequence learning with neural networks. In NIPS, pages 3104–3112, 2014.
[54] O. Vinyals, L. Kaiser, T. Koo, S. Petrov, I. Sutskever, and G. Hinton. Grammar as a foreign language. arXiv:1412.7449, 2014.
[55] J. R. Haritsa. The Picasso database query optimizer visualizer. PVLDB, 3(1-2):1517–1520, 2010.
[56] D. Silver et al. Mastering the game of go with deep neural networks and tree search. Nature, 529(7587):484–489, 2016.
[57] Y. Ma, X. Zhu, S. Zhang, R. Yang, W. Wang, ad D. Manocha. Traffic predicts Trajectory prediction for heterogeneous traffic agents. CoRR, abs/1811.02146, 2019.
[58] D. Wang, J. Zhang, W. Cao, J. Li, and Y. Zheng. When will you arrive? Estimating travel time based on deep neural networks. In AAAI, 2018.
[59] T. Kenter, A. Borisov, C. Van Gysel, M. Dehghani, M. de Rijke, and B. Mitra, "Neural networks for information retrieval," in Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval. ACM, pp. 1403–1406, 2017.
[60] Z. Wang, K. Fu, and J. Ye. Learning to estimate the travel time. In KDD, 2018.
[61] S. Wang, J. Cao, and P. S. Yu. Deep learning for Spatio-temporal data mining: A survey. CoRR, abs/1906.04928, 2019.
[62] C. Guo, C. S. Jensen, and B. Yang. Towards total traffic awareness. ACM SIGMOD Record, 43(3):18–23, 2014.
[63] D. R. Mould. Models for disease progression: New approaches and uses. Clinical Pharmacology & Therapeutics, 92(1):125–131, 2012.
[64] Z. Luo, S. Cai, J. Gao, M. Zhang, K. Y. Ngiam, G. Chen, and W. Lee. Adaptive, lightweight regularization tool for complex analytics. In ICDE, pages 485–496, 2018.
[65] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, et al., "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," IEEE Signal Processing Magazine, vol. 29, no. 6, pp. 82–97, 2012.

[66] A. Graves, A.-r. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in IEEE International Conf on Acoustics, Speech and Signal Processing, 2013, pp. 6645–6649.
[67] B. Hu, Z. Lu, H. Li, and Q. Chen, "Convolutional neural network architectures for matching natural language sentences," in NIPS, pp. 2042–2050, 2014.
[68] Y. Shen, X. He, J. Gao, L. Deng, and G. Mesnil, "Learning semantic representations using convolutional neural networks for web search," in IntnlConf on World Wide Web, pp. 373–374, 2014.
[69] L. Pang, Y. Lan, J. Guo, J. Xu, S. Wan, and X. Cheng, "Text matching as image recognition," in Thirtieth AAAI Conference on Artificial Intelligence, 2016.
[70] H. Zamani, M. Dehghani, W. B. Croft, E. Learned-Miller, and J. Kamps, "From neural re-ranking to neural ranking: Learning a sparse representation for inverted indexing," in 27th ACM International Conference on Information and Knowledge Management. ACM, 2018.
[71] S. MacAvaney, A. Yates, A. Cohan, and N. Goharian, "Cedr: Contextualized embeddings for document ranking," CoRR, 2019.
[72] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations," arXiv preprint arXiv:1802.05365, 2018.
[73] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," arXiv preprint arXiv:1810.04805, 2018.
[74] J. Guo, Y. Fan, Q. Ai, and W. B. Croft, "A deep relevance matching model for ad-hoc retrieval," in Proceedings of the 25th ACM IInternationalon Conference on Information and Knowledge Management. ACM, pp. 55–64, 2016.
[75] J. Worsham and J. Kalita, "Genre identification and the compositional effect of the genre in literature," in COLING, 2018, pp. 1963–1973.
[76] Y. Kim, "Convolutional neural networks for sentence classification," arXiv preprint arXiv:1408.5882, 2014.
[77] A. Conneau, H. Schwenk, L. Barrault, and Y. Lecun, "Very deep convolutional networks for text classification," in European ACL, vol. 1, 2017, pp. 1107–1116.
[78] M. Jiang, Y. Liang, X. Feng, X. Fan, Z. Pei, Y. Xue, and R. Guan, "Text classification based on deep belief network and softmax regression," Neural Computing and Applications, vol. 29, no. 1, pp. 61–70, 2018.
[79] G. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," Neural computation, vol. 18, no. 7, 2006.
[80] R. S. Sutton and A. G. Barto, Reinforcement learning: An introduction. MIT Press Cambridge, 1998, vol. 1, no. 1.
[81] P. Smolensky, "Information processing in dynamical systems: Foundations of harmony theory," Tech. Rep., 1986.
[82] D. Jurafsky and J. Martin, Speech & language processing. Pearson Education, 2000.
[83] Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin, "A neural probabilistic language model," J. of Machine Learning Research, vol. 3, 2003.
[84] W. De Mulder, S. Bethard, and M.-F. Moens, "A survey on the application of recurrent neural networks to statistical language modeling," Computer Speech & Language, vol. 30, no. 1, pp. 61–98, 2015.
[85] N.-Q. Pham, G. Kruszewski, and G. Boleda, "Convolutional neural network language models," in EMNLP, 2016, pp. 1153–1162.
[86] M. Lin, Q. Chen, and S. Yan, "Network in network," arXiv preprint arXiv:1312.4400, 2013.