A systematic survey on Information retrieval and Document ranking

Pragati Natwarlal Bhattad[1], Hritam Dutta[2], Dev Wadhwa[3],
Atharva Nitin Gundawar[4]

[1,2,3,4] Vellore Institute Of Technology, Vellore, Tamil Nadu, India

[1]pragatibhattad1610@gmail.com
[2]hritamloyola10@gmail.com
[3]devwadhwa01@gmail.com
[4]atharva.n.gundawar@gmail.com

under the guidance of
Prof. Saravanakumar Kandasamy
Associate Professor Grade 1
SCOPE
VIT, Vellore
ksaravanakumarvit@gmail.com

ABSTRACT

The constant emergence of large amounts of data - unprocessed - has made it difficult to extract information - useful data - for utilization in our applications. This is why various models and algorithms used for information retrieval and document ranking have attracted much attention in recent years for their potential to alleviate this problem. One of the major reasons that are limiting the utilization of proper algorithms for a given application is due to unawareness of the presence of different algorithms and models in this field. And thus, this paper proposes to solve this issue by presenting different algorithms with their own specific applications which can help a user identify the same for their application. This survey paper then presents the state-of-the-art research relating to different areas of these models, evaluating their strengths, weaknesses, and overall suitability for applications specific to them. Challenges that this field of information retrieval and document ranking face include expensive and complex all-to-all match retrievers, vocabulary mismatch, running the models on outdated data, the efficiency of the model, querying time, the inability of the model to handle a wide range of features, etc. Different challenges are presented in the paper and recommendations have been made for future research directions.

Keywords: Document Ranking, Information Retrieval, Transformers, SetRank, DeText, BERT, LSTM, RNN, COIL, TPRM.

1. INTRODUCTION

Information retrieval is a critical procedure for storing and retrieving data. It is defined as a totally automated process that examines a set of documents and returns a sorted document list that ought to be relevant to the user's requirements as indicated in the query.

Information retrieval and Document Ranking systems handle one of the most difficult challenges in knowledge management: swiftly discovering important information in vast databases and sorting the results by relevance. Organizations can profit right away from information retrieval. While it's critical to develop ways to capture tacit knowledge, the Information Query allows you to access data that's already in electronic form. Information retrieval software has progressed beyond simple search to include knowledge management activities like information dissemination. Web technologies and browser-based interfaces are used in information retrieval systems.

The first basic implementations of information retrieval models came into place during the late 1960s, researchers developed test systems and evaluation methodologies though large collections or databases. In the 1970s and 1980s, the exponential expansion of computing power aided research and large-scale testing of novel approaches for content indexing and query optimization. The National Library of Medicine (NLM) continued to work on its MEDLINE database. The growth of network technology in the 1990s allowed for more direct access to databases, and research into interface and query improvements, as well as research into massive "textual" databases, intensified. The National Library of Medicine has integrated its "Pubmed" interface into the MEDLINE system. The Text REtrieval Conference (TREC), an annual conference and series of workshops sponsored by the National Institute of Standards and Technology and the US Department of Defense's Disruptive Technology Office, began in 1992. The annual conference serves as a venue for "challenge assessments" of new approaches in the context of a "standardized task and/or data collecting." The emergence of the Web brought information querying and retrieval to the masses.

This section will discuss and describe the models present in the research papers we have chosen for our research. Every model is followed by its one liner explanation and then a paragraph describing the same.

1. DeText [1]: Non-exhaustive BERT implementation. Representation based structure: Instead of applying BERT to a concatenated string containing the query and document, the Qd structure produces query and document embeddings separately.
2. Contextualized Inverted List(COIL) [2]: Achieving independence from lexical matches by using soft matching : The creation of neural IR models is used to achieve the goal of soft matching all tokens. Despite the increased efficiency offered by deep LMs, there is still room for improvement by returning to lexical exact match systems with contextualized representations. This also allows an inverted list index to focus on only the subset of pages that have overlapping words with the query, which is a time-saving feature.
3. SetRank [4]: Precisely learns a permutation-invariant ranking model defined on document sets of any arbitrary size. The approach is permutation-invariant, which means it may be used straight to a set of texts without any preprocessing. Because self-attention networks, a prominent neural method used in machine learning tasks, were included in this paper's suggested model, it may be regarded as a deep model for IR.
4. Vanilla BERT [5]: BERT discards redundant attention weights on tokens with high document frequency (such as periods) at every step. Through their interactions, BERT aggregates document information to query token representations, yet derives query-independent representations for document tokens.
5. Vanilla Seq-to-seq [9]: Lacks BERT level precision but is far more data-efficient. The major advantage of this method, we feel, is that it allows one to leverage the model's latent knowledge (e.g., of semantics, linguistic connections, etc.) that has been polished by pretraining by "linking" fine-tuned latent representations of relevance to associated output "target words."
6. Aspect-based Document Similarity [7]: Unlike most RS which neglects the aspects which make two or more documents similar. Documents are connected in aspect-based document similarity based on the inner features that connect them. We use the title of the section in which a citation appears as a label for a document pair instead of citations for binary classification (i.e., similar and dissimilar). The aspect-based similarity of citing and cited publications is described in the section names of citations.
7. TPRM [8]: Improves effectiveness in modeling personalization signals, which had a low accuracy in most BERT implementations. User interest model, user-doc interest matching, query-doc semantic matching, and

personalized ranking are the four modules that assist accomplish the aforesaid levels of efficacy.

8. Local Self-Attention + Vanilla BERT [10]: Exponential decrease in memory cost and time required while enforcing the input documents. It operates with the help of a local self-attention system that examines fixed-size moving windows over the document terms, which helps to significantly reduce time and memory complexity. When the window size is modest in comparison to the duration of the sequence, the computer and memory needs are significantly reduced.
9. Hybrid Deep Fuzzy Hashing [11]: Handles security and other privacy features of input data. To improve the efficiency of information retrieval and data security in the distributed cloud environment, the mathematical formulation for the hashing technique and deep fuzzy for information retrieval are meticulously obtained. Implementing fuzzy with deep neural networks improves prediction accuracy via fuzzy logic and retrieval accuracy via neural network learning. To create a connection between the query and the database, a hashing algorithm is utilized.
10. BERT for document ranking [12]: BERT computations are expensive for document ranking. This causes a relevance-efficiency tradeoff for ad-hoc ranking. Despite the remarkable relevance of queries based on keywords, a long interference time is usually visible. Furthermore, the BERT model is still not able to achieve an NDCG relevance competitive to that of CEDR on ad-hoc ranking tasks.

As discussed above we can notice that every implementation is targeted to a very particular and peculiar problem, and every model has some scope of improvement in some capacity. The learnings from these models can be put into use to come up with better models which help solve most of the above-discussed problems.

The purpose of this article is to interpret and describe the significance of our findings in light of what was already known about and experiment on the research problem of information retrieval and document ranking. The problem being investigated and to explain any new understanding or insights that emerged as a result of our intensive study and research of models, implementations, and potential future work of improvising on the current systems.

As computing power improves and storage costs fall, the quantity of data we deal with daily increases dramatically. However, without the ability to retrieve and query the data, the information we collect is useless. To make sense of the data, information retrieval technologies are critical. Consider how difficult it would be to obtain information on the Internet if Google or other search engines were not available. Without information retrieval methods, information is not knowledge. Most organizations' data is dispersed among several different data repositories. Important repositories include file servers, groupware systems, relational databases, legacy systems, and even external sources such as the Internet. Text indexing and retrieval systems can index information in multiple data sources, allowing users to search against it. As a result, retrieval systems provide users internet access to information they might not be aware of, and they aren't needed to know or care where the data is kept. Users can use a single search to query all information that the administrator has considered suitable to index.

The paper is properly divided and described using headings, sub-headings, and so on. First of all, we have explained most of the relevant terms which might come in handy while reading this paper to better understand the meaning of each terminology. In the case of formulas, they have been explained and their mathematical scientific notation is given. Secondly, the architecture of the proposed model is described with a visual representation of the same to better understand the flow of data and generation of embeddings. Each paper is briefly explained after the above, each paper section has 2 subtitles from the paper and has been explained in detail. Following that, each author's contributions were discussed in depth. After that, all the evaluation metrics that have been discussed in the chosen papers, base papers, and other resources have been described with their mathematical scientific representations. This is followed by a conclusion and future work which concludes the paper and the research along with some ideas for what can be improved on in the upcoming papers. Finally, the paper ends with the references and citations for all the referred work and knowledge used in making this paper.

2. RESULTS AND GENERAL ARCHITECTURE

Each sequence to sequence model that is used for Information retrieval and Document ranking is unique in its way due to the different applications they are used for. We have discussed different models with their distinct features and applications, which can aid a user to choose the best-suited one for their work purposes. However, in almost all of them, a general base architecture is present which consists of different parts. The following figures represent a general model.

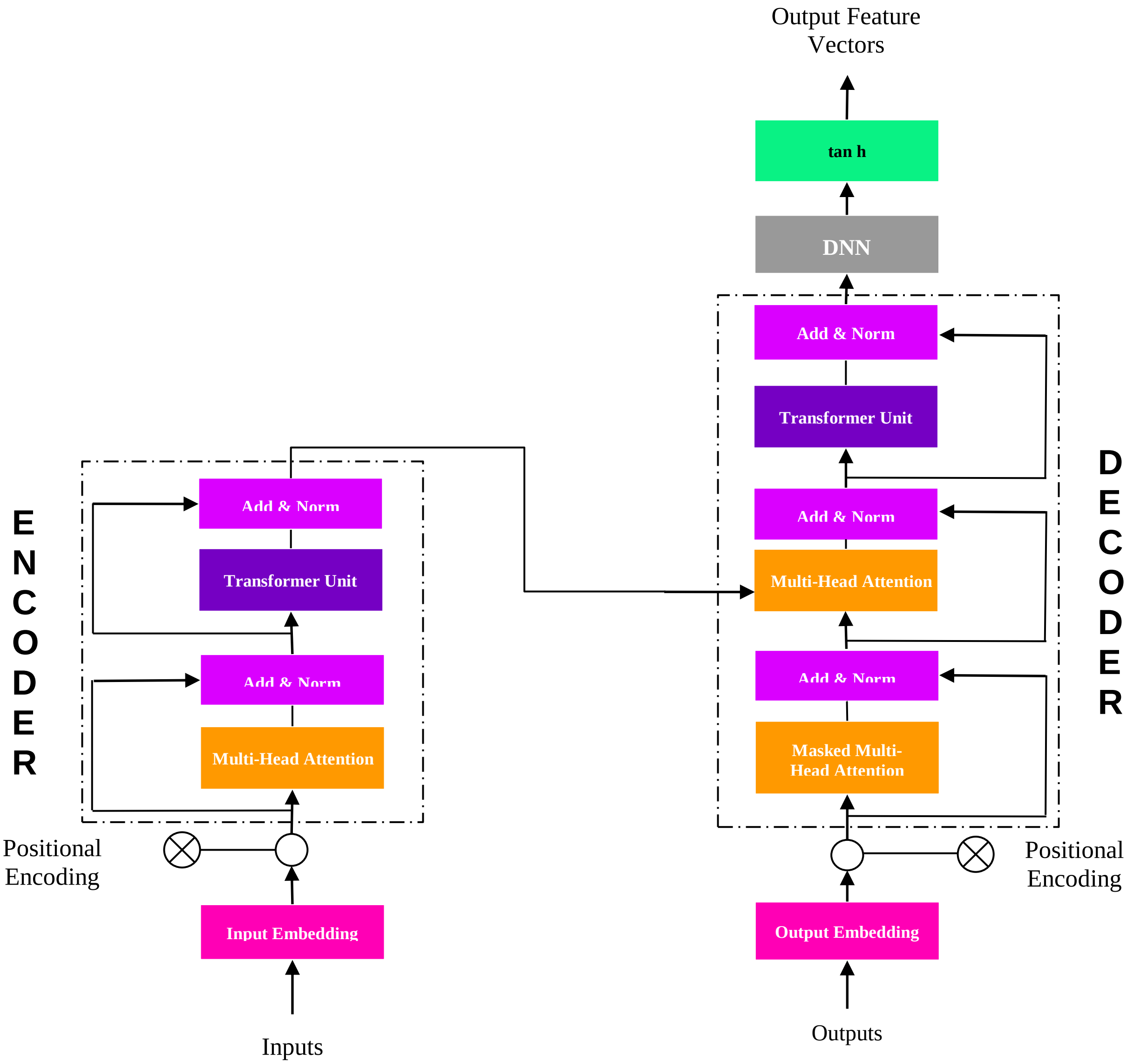

Figure 2.1: Generalized encoder-decoder architecture for a model

Figure 2.1 shows the entire model unit which will be used across time to take sequences as inputs and make them Recurrent. The first part on the left side is the Encoder which is the module that takes the document(s) as its input and stores the targeted information in its weights. It comprises the following modules: A. Multi-Head Attention block: Multi-threaded attention blocks help in capturing a higher amount of feature vectors that will be considered to calculate inference. B. The attention blocks are followed by Transformer units. These are the blocks that comprise 95% of the model. C. Between every unit there is an Add & Normalization layer which adds the output of the last sequence iteration and the current logits. The second part of the model is the Decoder which takes two inputs namely the logits/weights from the encoder and the input query. The logits/weights are responsible for saving user data from the input documents and the query is the question asked. It comprises the following modules: A. Multi-Head Attention block: Multi-threaded attention blocks help in capturing a higher amount of feature vectors that will be considered to calculate inference. B. The attention blocks are followed by Transformer units. These are the blocks that comprise 95% of the model. C. Between every unit there is an Add & Normalization layer which adds the output of the last sequence iteration and the current logits. D. Finally each output of the decoders is summed together in a basic dense neural network that uses tan-h as the output activation function.

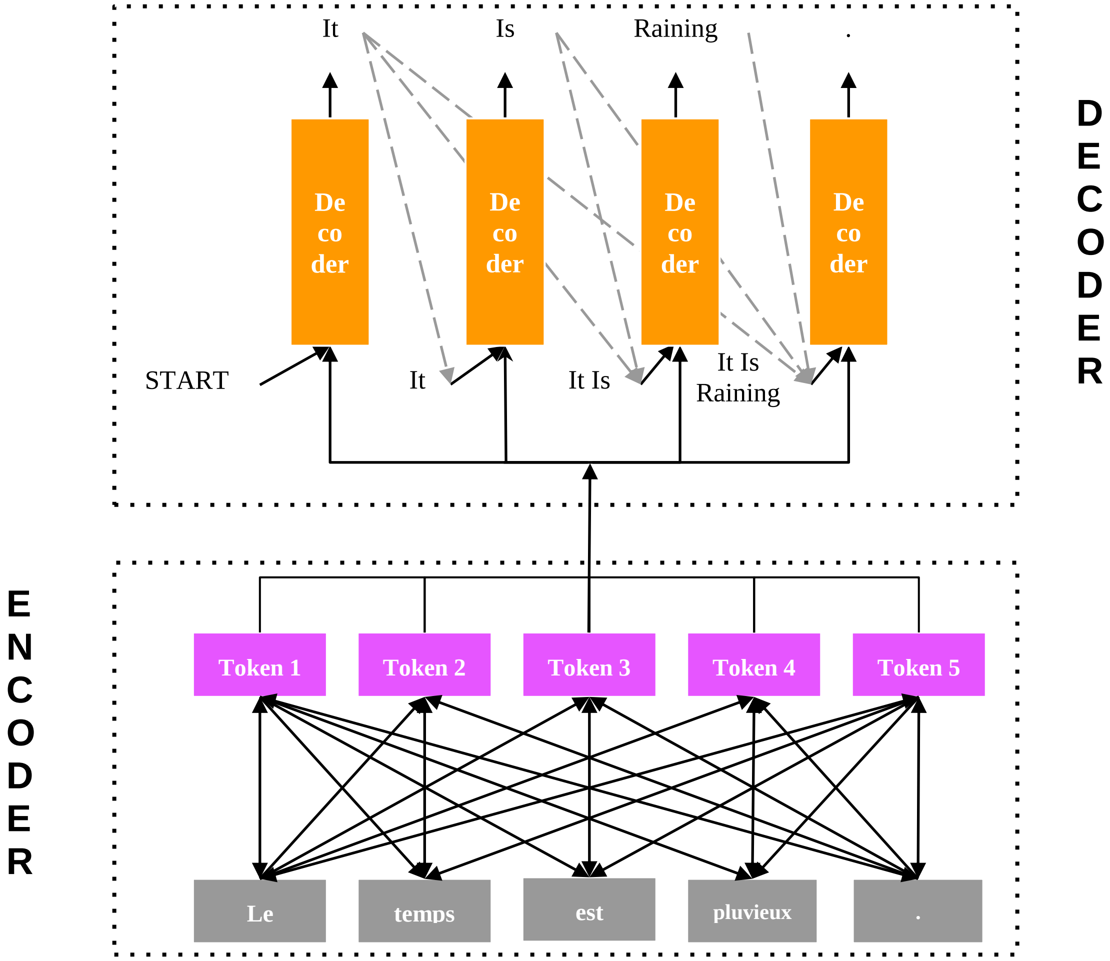

Figure 2.2: Working mechanism of the architecture on a set of inputs

Figure 2.2 is rolled out across time to better explain how it handles sequences. As you can see the blocks in gray denote the input document and the purple are the tokens or embeddings generated by the Encoder which will be passed to the Decoder to use as its knowledge base. The decoder then will take the input query and the knowledge base generated by the encoders to generate the output sequence.

3. SEQUENCE-TO-SEQUENCE MODELS

In this section, different sequence-to-sequence models are explored for document ranking and information retrieval, and their suitability for different applications is highlighted. Several pioneering works related to information retrieval and document ranking are discussed. Each of them has solved different problems and has innovative solutions with their architecture varying from simple to complex. A complete analysis of each of them has been made to use them for appropriate applications and guide the future development of such models.

3.1. Information Retrieval Models

3.1.1. Local Self-Attention Model

Document retrieval helps in finding relevant documents to user queries. When documents are retrieved, the time and memory cost might be high when using Transformers over a sequence of documents. A strategy wherein only the first n terms of the document are considered might lead to the formation of a biased system that retrieves lengthy documents. This paper deals with building a local self-attention to lower the compute and memory cost of attention over the whole document. The problem was chosen because the time and memory complexity of using self-attention over a sequence of length N is O(N^2), which makes it very inefficient to use over a long sequence of documents. A strategy wherein only the first n terms of a document are considered worked well when implemented for a short sequence of documents. But it retrieves longer documents because it only inspects the first n terms of any document. [59] discusses the difficulty in finding effective multi-level soft matches through neural models. K-NRM, a kernel-based neural model, was proposed for document ranking. Its kernel-guided embedding encodes a similarity metric build specifically for matching query words to document words and later provides effective multi-level soft matches. According to [60], depending on a first-stage ranker forms a dual problem such that the combination and interaction effects are not well known. Also, the stage ranker works as a filter, thus blocking the potential of neural models to find new relevant documents. [61] Furthermore, much attention has been paid to various neural ranking architectures, but not much attention has been paid to the term representations that are used as input to these models. It is seen that many present neural ranking architectures may benefit from the added context provided by contextualized language models. In [62], the vocabulary mismatch problem is addressed. Vocabulary mismatch occurs when different people name the same concept differently. A query generation method for document expansion that is based on the pointer-generator model is used for the same. As per [63], large transformer models routinely provide up-to-date outcomes on a variety of tasks. However, training these models can be expensive. Training such models is especially costly on long sequences. The proposed solution to tackle the problems involves implementing a local self-attention that considers fixed-size moving windows over the document terms. For each term, the terms in the same window need to be tended to. This reduces the time and memory complexity over a sequence of length N, from O(N^2) to O(Nn) wherein n is the window size.

The solution that is proposed is of a local self-attention that considers fixed-size moving windows over the document terms helps in the reduction of time and memory complexity by a considerable amount. If the window size is very small when compared to the length of the sequence, there is a significant reduction in computer and memory requirements. To combine the evidence from different parts of the document, a novel two-staged aggregation strategy is proposed. The first stage involves a local aggregation with a learned saturation function within a fixed window size and the second stage revolves around a global selection of top-t different important portions of the document and their corresponding signal aggregation. The proposed system, TKL, is more likely to retrieve longer documents than TK. Also, retrieval quality improves when TKL considers longer portions of documents. Thus, the longer the input document, the better the results. [64] TREC Deep Learning track dataset for document retrieval was used for training and testing purposes.

The three steps that are involved are:

1. Efficient Contextualization:
   Query embeddings in one window and document embeddings in multiple windows of size w and overlapped by o are contextualized.

2. Term Interaction and Kernels:
   TKL transforms each term interaction with kernel activations, which splits similarities into activations based on how close the values are to a certain range.

3. Topography Saturation and Scoring
   The top-t local maxima and their f immediate neighbors, by selecting 1 to f left and right values of the maxima are calculated. The local is then defined as saturation with region size r, so that term matches are not counted twice.

The proposed system, TKL, is more likely to retrieve longer documents than TK. Also, retrieval quality improves when TKL considers longer portions of documents. Thus, the longer the input document, the better the results. However, when working with shorter documents, TKL and TK have the same results. Thus TKL might not be needed in some cases.

3.1.2. COIL: Contextualized Inverted List Model

While retrieving correct data based on a query is an important factor in determining how good the model is, the efficiency and speed of the model also are valid factors for the same. The loss in the computation efficiency of the exact match system because of soft semantic matching all query document terms is what this paper focuses on [2].

Generally, information retrieval systems such as BM25 are dependent on the exact lexical match and that conducts efficient searching with inverted list index, but the recent neural IR models shifts towards soft semantic matching all query document terms, but eventually losing the computation efficiency of exact match systems as a disadvantage. The idea of soft matching all tokens is done through the development of neural IR models. On having the improved efficiency brought by deep LMs but there is still a scope of gaining more efficiency by contextualized representations back to lexical exact match systems. This also allows the search to focus on only the subset of documents that have overlapping terms with a query, which can be done efficiently with an inverted list index. At the same time, the use of dense contextual token representations allows the model to deal with semantic incompatibility, which has been a long-standing problem in classical lexical systems.

In the past, there were four different approaches and improvisations for the same cover-up problem

1. Bag-of-words (BOW) information retrieval (IR) systems that are widely popular such as BM25 depend on exact lexical match 2 between query and document terms.
2. Later the study in neural IR leads to a different approach that computes between n all query and document terms to model intricate matching
3. Further introduction of the contextual representation of deep linguistic models (LM) further focuses on semantic incompatibility, i.e. the same word can refer to different meanings.
4. Re-ranking LM in generating context-based token representation and achieving state of the art in text classification with huge performance improvements

To gain more insight on how this problem was tackled in the past, many other papers with similar insights have been referred to in the paper [2]. [18] suggests analyzing the capacity of the dual encoders to encode the document into dense low dimensional vectors and scoring each of them by the inner product query method relative to sparse bag-of-words models and attentional neural networks. While [19] emphasizes Accelerating Large-Scale Inference with Anisotropic Vector Quantization which implies a score-aware quantization loss function that can work under any weighting function of the inner product and regardless of whether the data points vary in the norm, in turn, increasing the efficiency of the loss function. [20] explains a Context-Aware Sentence/Passage Term Importance Estimation for First Stage Retrieval. This paper proposes a solution to handle the importance of a term when the term frequency distribution is flat. In turn as a solution, a Deep Contextualized Term Weighting framework can be implemented to map BERT's contextualized text representations to context-aware term weights for sentences and passages. [21] presents words, phrases, and their compositions with a distributed representation. This paper proposes an extension to Skip Gram and studies high-quality distributed vector representation. This can be achieved by reducing frequently used words by learning a more regular representation of words. After analyzing more of these models, the following model was created.

It works on a new lexical matching scheme that uses vector similarities between query-document overlapping term contextualized representations to replace heuristic scoring used in classical systems. It processes documents with deep LM offline and produces representations for each document token. In addition, the representations are grouped by their surface tokens in an inverted list. At search time, each query token is used to search its reverse list and calculate the similarity of the vector to the document vector stored in the reverse list as a match score

The following are the algorithms that this model uses:

1. Inverted list indexing - The inverted list tracks back from a term to a collection of documents where the term appears. As an advantage retriever only needs to traverse the subset of documents in query terms' inverted lists.
2. Contextualized Exact Lexical Match Scoring - A specific derived function can be used to evaluate similarities between exactly matched query document token pairs.

Contextualization improves the capability of modeling intricate matching patterns. COIL-full's when the dimensions are lowered leads to a faster result than the DPR system. It can analyze its own embedding space and can encode and measure the semantic similarity of the token in different contexts. However, the possibility of Vocabulary mismatch still persists. Expensive and complex all-to-all match retrievers.

3.1.3. Hybrid Deep Fuzzy Hashing Algorithm based Model

Information management systems are always in demand by a user to access the internet-based services, network mediums, electronic libraries, and recently advanced search engines. Therefore, an efficient information retrieval system that also caters to the need of maintaining the security and privacy of data is the need of the hour. Numerous information retrieval models have emerged for this purpose. But, each model has its advantages and disadvantages. In order to overcome the limitations of the existing models such as ranking-based information retrieval model, multimodal retrieval system, cluster-based information retrieval model, etc., a hybrid data processing model is required. A problem that emerged in the information management system is the cost involved in the process. The cost of data management for storage and retrieval is huge. Hence, cloud computing evolved as a cost-effective method of information management. But, the cloud has some limitations such as communication failure, data loss, data traffic surveillance, and other potential risks in terms of privacy and security. Also, cloud computing is the most common method for information management but it has few limitations pertaining to data privacy.

[65] Yongjun. et.al. described the problems encountered due to the natural language interface in the bibliographic information retrieval system. As the database management system faces difficulty in organizing natural language data, the retrieval of information using natural language becomes difficult. In order to overcome these issues, an interface is proposed by the researchers which help the user to search bibliographic data using natural language. [66] Sidali Hocine Farhi. et.al. introduced a graph-based information retrieval system that is widely adopted in many applications. Keeping the emphasis on the graph-based system, the proposed bibliographic information system is developed which processes the queries as text and retrieves information through the interface. [67] Andrei. et.al. proposed a Ranking based information retrieval model in their research work. Based on document description and term frequency model, ranks are allocated considering the user request. The proposed research uses modified genetic algorithms and provides relevant information with minimum stagnation. The structural complexity in conventional information retrieval is reduced in the proposed approach using ranking models and genetic-based map criteria. The limitation of the model is that the proposed system failed in processing natural language requests. [68] Jianchang Lai. et.al. proposed a private information retrieval model which allows the user to retrieve the information from the database based on user preference. In the case of conventional private information models, each data is required to be published with a description which leads to information leakage. In order to minimize such information leakage, an attribute-based information retrieval model is proposed in the said research work. The proposed work attains better data privacy which does not reveal any information about the data. The complexity of this model is its limitation as it is difficult to describe the data which is present in a large data set using this model. [69] Jennifer. et.al. described the problems faced in multi-cloud environment information processing. Multi-cloud provides better consistency and diminishes the severity. But information maintenance in a multi-cloud environment is difficult due to its interface, service renders, and technologies. Information maintenance such as store, secure, and retrieve processes need a highly reliable and flexible system. The proposed model overcomes the limitation in traditional cloud-based information processing systems through cryptography integrated computational intelligence which is widely adaptable for multi-cloud models. But, the security of data is a major issue in multi-model data processing systems. The researchers concluded that though the performance of the multimodal information system is better, it is essential to improve the features of the system. [11] The proposed solution to tackle such problems is a hybrid deep fuzzy hashing algorithm model.

The proposed solution in this research paper is a hybrid deep fuzzy hashing algorithm model. The mathematical formulation for the hashing approach and deep fuzzy for information retrieval are methodically obtained to enhance the efficiency of information retrieval and security of data in the distributed cloud environment. Implementing fuzzy with deep neural networks enhances prediction accuracy through fuzzy logic and retrieval accuracy through the learning ability of neural networks. The hashing function is used to establish a relationship between the query and the database. The proposed hybrid deep fuzzy hashing algorithm for distributed cloud environments is verified experimentally in CloudSim and the results are observed. [70] KDD Cup 2004 Database is used in the experimentation which comprises 50000 training examples, 100000 test examples, and 78 numerical attributes.

The classification and retrieval efficiency of the proposed model is measured in terms of the following:
i) Specificity
ii) Sensitivity

To determine the efficiency of the proposed model, it is compared with the Support vector machine-based information retrieval system and Deep neural network-based information retrieval system. A hybrid combination of deep learning and fuzzy hashing algorithm enhances the information and data administration in a cloud environment. The proposed model is verified experimentally and compared with the existing Support vector machine-based information retrieval system and Deep neural network-based information retrieval system. The proposed model obtains 97.6% retrieval efficiency which is regarded as an exceptional improvement in information retrieval systems. As stated by the authors, future work may involve the use of optimization models to reduce the number of features in the feature selection process.

The proposed model achieves 97.6% retrieval efficiency which is considered a remarkable improvement in information retrieval systems. However, the limitation of the proposed solution is handling a wide range of features.

3.2. Document Ranking models

3.2.1. Data-Efficient Pretrained Sequence-to-Sequence Model using BM25

The sequence-to-sequence model seems to be far more data-efficient: the approach used in this work [6] shines in a data-poor regime.

Since it is not practical to apply inferences to all documents in a corpus related to a query, these techniques are commonly used to reorder a candidate list. In a typical end-to-end system, these candidates are obtained from the results of a keyword search based upon a "classic" IR scoring function such as BM25. This leads to the standard multi-stage keyword retrieval pipeline architecture, followed by re-classification using one or more machine learning models. The sequence-by-sequence model appears to be much more data-efficient. In a data-rich regimen with many training examples, this method may outperform an encoder-only approach based purely on classification. However, the sequence-by-sequence model appears to be much more data-efficient - our approach shines in a data-sparse regime and outperforms BERT with limited training examples. The main advantage of this approach, in our opinion, is that by "connecting" latent representations of finely tuned relevance with related output "target words", the latent knowledge of the model is obtained (for example, about semantics, linguistic relationships, etc.) that has been developed through prior sharpening training. A straightforward formulation of classification is to turn the task into a classification problem and then order the candidate elements to be classified according to the probability that each element belongs to the desired class. Applied to the problem of document precedence in information retrieval - when queried, the system's job is to return a document classification list from a large corpus that maximizes some classification metrics such as average precision or nDCG - the formulation simpler is to provide a classifier that estimates the probability that each document belongs to the "relevant" class and then ranks all candidates according to these estimates. Deep Transformer models pre-trained for language modeling purposes, using BERT as an example, have proven very effective in a variety of sequence labeling and classification tasks in NLP.

[38] Deep Transformer models that are pre-trained for language modeling purposes, using the BERT example, have been shown to be very effective in a variety of NLP sequence tagging and classification tasks. [39] In a typical end-to-end system, these candidates come from the results of a keyword search based on a "classic" IR scoring function such as BM25. [40] T5 bears the work of tokenizing sequences using the SentencePiece model, which could divide a word into sub-words. We choose target words ("true" and "false") that are represented as individual tokens; thus, each class is represented by a single logit. [41, 42] leads to the standard multi-stage keyword retrieval pipeline architecture, followed by a reordering using one or more machine learning models.

This work proposes a novel adaptation of a previously trained sequence-by-sequence model to the document classification task. This approach differs fundamentally from a generally accepted classification-based classification formulation that relies on previously trained converter architectures for encoders only. like BERT. This thesis depicts how a sequence-by-sequence model can be trained to generate relevant tags such as "target words" and how the underlying logits of these target words can be understood as relevancy probabilities for ranking.

Few relevant concepts and algorithms that have been used in the current work are:

1. Greedy decoding during inference.
2. This re-ranking method is based on T5, a sequence-to-sequence model that uses a masked language modeling target similar to that of BERT to pre-train its encoder-decoder architecture.

It should be noted that T5 tokenizes sequences using the SentencePiece model, which can divide a word into sub-words. This model chooses target words ("true" and "false") that are represented as individual tokens; Therefore, each class is represented by a single logit. Instead, this model has an adaptation. At inference time, this model applies a softmax only to the logits of the "true" and "false" tokens to calculate the probabilities for each query and document pair (in a reorder configuration). Therefore, this model reorders the documents according to the probabilities associated with the "true" token.

This model outperforms a classification-based approach, especially in the data-poor regime with limited training data. This approach significantly outperforms an encoder-only model in a data-poor regime. This approach is more data-efficient than BERT. T5 significantly outperforms BERT when fine-tuned with few training examples. However, the exact way the model is exploiting knowledge from its ability to generate fluent natural language text is still unclear. This approach is much more suitable for a data-poor regime, as it yields outstanding results. There are few reservations for this to do well in a data-rich regime. This work's experiments are inconclusive regarding the importance of having a polarity scale in the low-data regime.

3.2.2. ELECTRA or Aspect-based Document Similarity Model

The traditional document similarity measures do not consider in what aspects two documents are similar. This limits the granularity of applications such as recommendation systems that rely on document similarity. When user feedback is scarce or unavailable, content-based approaches and appropriate document similarity measures are used (Beel et al., 2016) [37]. RS recommends candidate documents based on whether they are similar or dissimilar to seed documents. This rough assessment of similarity (similar or not) ignores the many facets that can make two documents similar. Regarding the concept of similarity in general, Goodman (1972), and Bar et al. (2011) even argue that similarity is a poorly defined idea unless it can be said which aspect refers to the similarity.
[39, 40, 41] In RS for scientific articles, similarities often affect some aspects of the research presented, for example, methods, results (Huang et al., 2020) [38, 42].

Most RSs are based on a measure of similarity between seeds and the k most similar target documents (a). Aspects that make two or more documents similar are ignored. In the similarity of documents based on the aspect (b), related documents according to internal aspects that connect them (a1 or a2). Multi-label scenarios and focus on scientific literature rather than general literature (Wikipedia article). Similar to the work of Jiang et al. (2019) and Cohan et al. (2020), we use quotes as training signals. [37, 38] Instead of using citations for binary (i.e. similar and different) classification, we include the headings of the sections where the citations occur, as labels for a pair of documents. The title of the citation section describes the similarities based on the aspect of the citation and the cited article. Our data set comes from ACL Anthology (Bird et al., 2008) and CORD19 (Wang et al., 2020). [40, 41]

When recommending literature, content and user information are the predominant dimension to consider (Beel et al., 2016). Chan et al. (2018) examine aspect-based document similarity as a segmentation task rather than a classification task. Huang et al. (2020) apply the same segmentation approach to the CORD19 corpus (Wang et al., 2020). Kobayashiet al. (2018) takes a similar approach to citation recommendations. [42, 43]

Experiments investigate the Transformer language model (Vaswani et al., 2017). BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019), XLNet (Yang et al., 2019), and ELECTRA (Clark et al., 2020) improve many NLP tasks, such as natural language inference (Bowman et al., 2015; Williams et al., 2018) and semantic textual similarity (Cer et al., 2017). Reimers and Gurevych (2019) demonstrated how the BERT model can be combined in a Siamese network (Bromley et al., 1993) to produce comparable embodiments using cosine similarity. Adikari et al. (2019) and Ostendorff et al. (2019) explore BERT for the classification of individual documents with respect to sentiment or topic. Beltagy et al. (2019) and Cohan et al. (2020) studied domain-specific transformers for NLP assignments on scientific papers. [45, 46, 47]

ELECTRA and RoBERTa achieve higher F1-scores than Covid-BERT and XLNet. Even though Covid-BERT is fine-tuned on CORD-19 its performance yields a 0.818 micro-F1. However, SciBERT presents the highest scores with a large margin, followed by Covid-BERT, XLNet, and BERT. The lower performers are RoBERTa (0.626 micro-F1) and ELECTRA (0.616 micro-F1)

3.2.3. Context-Aware Document Ranking model by Contrastive Learning of User Behavior Sequence

The main aim of this document is to obtain a better representation of user behavior that can accommodate the high variability.

Existing studies explore the sequence of user behavior to improve query suggestions or document ranking. However, the sequence is often seen as a precise and precise signal that reflects user behavior. This does not take into account the variability of user behavior as user requests for the same intent vary. Search engines have evolved from one-time searches to successive search interactions with users. [50] To meet complex information needs, users will issue a sequence of questions, check and interact with multiple results. User behavior history or interaction history within a session is known to be very useful for understanding user information needs and for classifying documents. Various studies have leveraged user behavior data by analyzing search logs and finding that a user's search history provides useful information for understanding user intent during a search session. [51]

The order of user behavior is seen as a defined and precise order. That is, observed sequences are used as positive samples and unseen sequences are not used or viewed as negative samples. [52] This strict view does not reflect the flexible nature of user behavior within a session. Different approaches and improvisations were followed to cover the same problem

1. Hierarchical neural structures with RNNs [53] were used to map these sequences
2. Attention mechanisms to better represent sessions and capture search behavior at the user level
3. Shared learning of suggestions Query and document classification can improve model performance on Duty. [54, 55]
4. Large-scale pre-trained language models, such as BERT, yield strong results for these tasks

Here you are working on a new lexical matching scheme that uses vector similarities between conceptual-contextualized representations of overlapping query documents to replace the heuristic evaluation used in classical systems. Process documents with Deep LM offline and create representations for each document token surface token in inverted lists. [56] At search time, each query token is used to find its own inverted list and calculate vector similarity to document vectors stored in the inverted list as matching scores.
COCA performs better, demonstrating its effectiveness in modeling user behavior sequences through contrastive learning. In general, the contextual document classification model performs better than the ad hoc classification model. For example, on the AOL dataset, the weak contextual model of MNSRF can still outperform the strong ad hoc ranking model of KNRM. [57, 58] This shows that modeling the order of user behavior is useful for understanding user intent and improving ranking results.

COCA achieves the best results, which demonstrates its effectiveness in modeling user behavior sequence through contrastive learning. In general, the context-aware document ranking models perform better than ad-hoc ranking models. For example, on the AOL dataset, the weak contextualized model M-NSRF can still outperform the strong ad-hoc ranking model KNRM. This indicates that modeling user behavior sequence is beneficial for understanding user intent and improving the ranking results. However, the possibility of Vocabulary mismatch still persists. Expensive and complex all-to-all match retrievers.

3.3. BERT and its Adaptations

3.3.1. Vanilla BERT

The understanding of BERT’s internal mechanism is not enough, even though BERT has shown its efficiency in several IR-related tasks. This paper [5] studies the evolution of attention distribution and aims at understanding BERT in a more detailed manner.

This paper studies the evolution of attention distribution. BERT discards redundant attention weights on tokens with high document frequency at every step. This may lead to a potential threat to the model’s robustness and should thereby be considered for future research purposes. By studying BERT models interactions between query and document, it has been indicated that it is possible to transform BERT into a more efficient representation-focused model. These findings might aid in better clarity of the ranking process by BERT and may inspire future improvement. After being pre-trained on a bigger corpus and finely tuned on supervised data, BERT can achieve promising results in ranking tasks. On the MS MARCO Passage Ranking leaderboard, BERT is adopted by the most exceptional performers. Attention largely does not impart meaningful "explanations" and researchers have been trying to analyze BERT, which is based solely on attention mechanisms. This research paper is of some significance as the elucidation of BERT in the ranking task has not been fully studied. Several studies noticed a surprisingly substantial amount of attention focusing on "[CLS]", "[SEP]" and periods, which is not fully understood. A research paper designed a representation-focused BERT ranker. But due to its poor performance, they suggested that BERT shouldn't be used as a representative model.

[33, 34] have analyzed the usage of BERT in document ranking, via a common procedure, which is to build the model input by concatenating the query and document text. [35] has found that some BERT attention heads agree well with linguistic notions of syntax and coreference. This paper analyzes the working mechanism of BERT. [36] found that the deeper layers produce more context-specific representations. In particular, they found that the contextualized representations of all words in any layer of the contextualization model are not isotropic. [37] studies the performance and behavior of BERT in ranking tasks and explores various ways to exploit the preformed BERT and refine it on two ranking tasks. The paper also demonstrates BERT to be a strong interaction-focused seq2seq matching model. [38] states BERT is designed to pre-train deep bi-directional representations from the unlabeled text by conditioning the left and right context on all layers together. As a result, the pre-trained BERT model can be refined with just one additional output layer to create advanced models for a wide variety of tasks.

To address the problem of insufficient understanding, the document analyzes 2 metrics. First, the development of the distribution of attention is considered. It shows that BERT discards redundant attention weights in tokens with a high document frequency (such as points) at each step. Second, it examines how BERT models query-document interactions, showing that BERT aggregates document information to query token representations through its interactions, but extracts independent query representations for document tokens.

Few relevant concepts and algorithms that have been used in the current work are:

1. Attention Distribution.
2. BERT model interactions between query and document.
3. Attribution techniques to study token importance in different layers.

The differences between the work in this paper and previous similar studies have been described. First, this paper generalizes the conclusion about "[CLS]" and the periods and attributes them to their high frequency of documents. Furthermore, such a demonstration of behavior can affect the robustness of the model. Secondly, we see that document token representations are largely independent of queries, showing great potential for improving BERT efficiency. This result differs somewhat from that suggested in Method 2. This article also examines interaction behavior at different layers, most of which have yet to be examined.

The paper generalizes the conclusion to "[CLS]" and periods, to somehow counter their high document frequency. Representations of document tokens are largely query-independent, thereby revealing the chance to transform BERT to a representation-focused model. The performance of BERT is compared when its attention matrix is masked in different ways to gauge the importance of interactions. However, one needs to be careful with tokens with high document frequency, as BERT discards redundant attention weights and periods due to their increased document frequencies. It isn't as efficient. It can be transformed into a more efficient representation-based model.

3.3.2. DeText - A Deep Text Ranking Framework with BERT and CLSM

BERT is a model that learns contextual embedding. But, this model uses exhaustive iteration over each query word with each document word which is very ineffective. [1] deals with building an efficient BERT-based ranking model. Document ranking is an important part of a search system and these systems deal with a large amount of Natural language. Deep learning-based NLP models have been made for ranking this data. BERT is the most advanced of them. The only downside is that it uses exhaustive iteration for learning and processing contextual embedding. This process includes iterating over each query word and comparing it with each document word which is time-consuming. Querying large amounts of data is a huge problem faced by search systems. To create an algorithm that is better than the previous BERT model to perform this function is necessary to improve efficiency. There are many other state-of-the-art deep NLP components in addition to BERT.

The solution proposed in [1] is a Representation-based structure: creates query and document embeddings independently instead of applying BERT to a concatenated string of the query and document. This model solution was further extended into a general ranking framework, DeText (Deep Text Ranking Framework). The paper mentions 2 categories of deep NLP based ranking models

1. Representation-based and
2. Interaction-based

Representation-based models are those that learn independent additions for the query and the document.

- DSSM averages word embeds as query/document embeds.
- CLSM / LSTMRNN encodes word order information using CNN / LSTM, respectively. All these three jobs assume that there is only one field on the document page and the document qualification is the cosine similarity qualification of the document query/embed.
- NRMF adds more fields on the document page and works better.

One of the major weaknesses of representation-based networks is the inability to capture local lexical matching. To overcome this, interaction-based models compare every part of the query against every part of the document.

a. In DRMM, a cosine match is calculated for each word embedded in the query and each word embedded in the document.
b. KNRM and ConvKNRM extended DRMM with kernel pooling and pairwise n-gram matching, respectively.
c. BERT has recently shown superior performance in the rankings.

In this model, the input text data has a source text and a target text. The source text can be a query or a user profile. The target text can be documented. The proposed architecture has 6 components:

1. Text Embedding Layer: Text Embedding Layer: The sequence of text tokens is transformed into an embedding matrix E.
2. Token Embedding Layer: This is a representation-based model. Structure and embedding are extracted independently for each text field.
3. Interaction Layer: The interaction between source and destination only occurs after the text insert is made, which is the main difference between representation-based methods and interaction-based methods.
4. Traditional Feature Processing: Existing features, such as personalization features, social networking features, user behavior features, integrate with NLP using standard normalization and per-item scaling
5. Multilayer-Perceptron Layer: MPL is used to print the final document. The hidden layer in MLP is able to extract nonlinear correlations from deep features and traditional features.
6. Learning-to-rank Layer: The final level is learning levels to rank which accept multiple target scores as input. Systems where only relative position matters are the main systems where LTR can be used.

BERT is one of the best models out there. This model surpasses BERT and is one of the most progressing and promising models.

This paper successfully deals with building an efficient BERT-based ranking model. BERT uses exhaustive iteration over each query word with each document word which is very ineffective. Instead, this model improves efficiency by creating query and document embeddings independently instead of applying BERT to a concatenated string of the query and document. Pretraining a BERT model on in-domain data can maintain the same level of relevance performance, while the current model significantly reduces computation. Deep ranking models were used in this paper which can achieve better performance than the production models. However, BERT has been a very strong model in itself. This paper has removed the flaws BERT had.

3.3.3. Composite Re-Ranking with BERT

The development of new transformer architectures (e.g., BERT) and pre-training techniques gave new life to the neural ranking studies using deep contextual models, leading to a higher relevance score in the top □ document search. Even though there is a striking relevance of the keyword-based queries using transformer-based ranking models, the long inference time is still a problem. In order to address this limitation, recent studies examined the efficiency of transformer-based models during model inference. Additionally, in order to decrease the complexity associated with the transformer computation, simplified BERT-based architecture was previously proposed to reduce the load of jointly encoding query-document pairs, including the twin encoder-based rankers that individually encode the query and the document. Moreover, despite the enhanced efficiency, the previously mentioned models are still not able to deliver an NDCG relevance competitive to CEDR on ad-hoc ranking tasks. To overcome these obstacles, the researchers in this study aim to develop an integrated re-ranking scheme, which they termed as BECR (BERT-based Composite Re-Ranking).

[71] Not much attention has been given to the term representations that are used as input to neural ranking models. The authors examine how two pre-trained contextualized language models (ELMo and BERT) can be used for ad-hoc document ranking. The study also reported the highest relevance for the popular ad-hoc datasets (ClueWeb and TREC Disk 4&5). [72] In the study, the researchers propose a new Conformer layer to reduce the memory complexity of the Transformer layers with respect to the input sequence length. The transformer-kernel (TK) model for natural language queries uses a lightweight BERT-like architecture with only 2 encoder layers, thus enhancing the online performance. [73] Although being effective, the ranking models based on the fine-tuning language models increase computational cost. [74] It is difficult to prevent the server from discovering embedding-based semantic features and inferring privacy-sensitive information because of the fact that neural ranking adds more complexity in score computation. Evaluation of the significant leakages in interaction-based neural ranking is done and then countermeasures to relieve such leakages are analyzed. [75] Interpretability of learning-to-rank models is a crucial research area. Recent advances in interpretable ranking models have largely focused on generating post-hoc explanations for existing black-box classification models, while the alternative possibility of building an inherently interpretable ranking model with a transparent and self-explanatory structure remains unexplored. The proposed solution tackles the mentioned problems.

The proposed solution in this paper is to implement transformed-based composite models together with traditional signals in order to furnish substantially enhanced efficiency and relevance in ad-hoc re-ranking. BECR further leverages embedding approximation based on locality-sensitive hashing and thus improves the efficiency of the model. The proposed framework involves token encoding, online composite re-ranking, storage cost, and reduction strategies.

The three major steps involved are:

1. Training:
   During training time, a set of training queries with their matched documents is sent as input to the BECR model to learn novel parameters.

2. Indexing:
   During indexing time, a set of unigrams and skip-bi-grams are selected and then fed to the trained model to pre-compute token embeddings. After compression, the results obtained are stored in a key-value store for online access.

3. Online Inference:
   During inference time, the neural representations of a query captured by a set of terms are considered. For each query, a semantic token set called T that consists of unigram or word pair tokens related to the query are also derived.

BECR delivers fast response time on affordable computing platforms. With token encoding, the query representations are effectively approximated by BECR. It also makes proper use of deep contextual and lexical matching features, allowing for strong ad-hoc ranking performance.

BECR delivers fast response time on affordable computing platforms. With token encoding, the query representations are effectively approximated by BECR. It also makes proper use of deep contextual and lexical matching features, allowing for strong ad-hoc ranking performance. However, BECR performs slightly worse than CEDR-KNRM but still outperforms BERT for Robust04, which is a relatively small dataset.

3.4. A combination of document ranking and information retrieval models

3.4.1. SetRank - Learning a Permutation-Invariant Ranking Model

Failure of univariate scoring functions to model interactions between documents; or multivariate scoring functions that sacrifice the requirement of permutation invariance. This paper [4], also inspired by the work of Set Transformer, was written to address the decrease in performance of previous models.

Previous studies of learning ranking models have shown that univariate scoring functions that score each individual document do not model interactions between documents. Multivariate scoring functions that score documents one at a time, ultimately sacrificing the need for permutation invariance. This paper [4] has addressed the above problems by proposing a neural range learning model called SetRank, which accurately learns a permutation invariant ranking model defined for document sets of any size. To cope with the decrease in performance of previous models. This article suggests developing a multivariate ranking model called SetRank, whose input is a set of documents of any size and whose output is a permutation over the set, making full use of the self-attention mechanism. If you compare existing models that use multivariate ranking functions, SetRank is more natural for document ranking and easier to use for parallel computing. It is also possible to include multiple document rankings as initial rankings. Conventional learning-to-rank models are generally based on the probability ranking principle (PRP). The strength of PRP-based learning models for classifying is limited. Independent assessment paradigms prevent traditional range learning models from modeling interactions between documents and capturing information from the local context. The PRP works document by document, while the ranking results are evaluated request by request. Annotation studies on the relevance of reference documents depict that information from other documents in the same ranking list can influence an annotator's decision on the present document, which in turn could challenge the basic hypothesis that the relevance of each document is modeled independently for a single information request. All existing multivariate evaluation approaches react sensitively to the order of input documents, thereby violating the permutation invariance requirement for the ranking model.

[28] proposes to encode the local context of an initial ranking with a recurrent neural network, and then to re-rank the documents, according to the integration of the latent context of all documents. [29, 30] suggests a slate optimization framework that directly predicts the ranking of a list of documents by mutually considering their attributes together. [28] further formalizes the scheme and proposes a multivariate scoring framework that is in accordance with deep neural networks. It is mainly based on the intricate interactions between top results from the deep neural networks. [31] suggests using a self-attention network to aggregate cross-documents information, in response to the problem under view. It satisfies the permutation-equivariant requirement and can produce scores for document sets of varying sizes. [32] proposes to build a list-wise re-ranking framework that personalizes recommendation results by jointly considering multiple items in the same ranked list. The proposed model can easily be used as a follow-up module after any classification algorithm by directly using the existing ranking feature vectors.

In this work, the authors proposed a new learning-to-rank model on the multivariate scoring paradigm. The method used is permutation invariant and can be applied directly to a set of documents with or without preprocessing. The model proposed in this article can be considered a deep IR model because it used self-attention networks, a popular neural technique used in machine learning tasks. The focus is on a general learning-to-rank task assuming that the feature representation for each pair of query documents has been calculated in advance. This model framework also supports learning ranking models and characteristics together.

The following concepts and algorithms were used in the paper to complete off the proposed work:

1. Ranking model satisfying permutation invariance requirements and cross-document interactions.
2. Learning-to-rank model using multivariate scoring functions.
3. Deep neural networks for document ranking.
4. Self-attention mechanism.

The scoring function in SetRank is designed as a multivariate mapping from a document set to a permutation. Efficiently captures local context information. Naturally involves (multiple) initial rankings. High accuracy in ranking. Robust to input noise. It can involve zero, one, or multiple initial document rankings as inputs, through adding one or more ordinal embeddings. However, there could be more work done on multivariate scoring functions for greater efficiency, even though SetRank has outperformed the traditional learning-to-rank models.

3.4.2. TPRM - A Topic-based Personalized Ranking Model for Web Search

Despite notable improvements in ranking performance, pre-trained, context-focused BERT-based rankings were less effective at modeling personalization signals, such as user clicks.

In a custom classification system, documents are classified based on user interests and specified queries. Specifically, for both q user-specific requests u and d documents, the specific classification system aims to calculate the relevance scores (u, q, d) based on user interest, query representation, and documentary representation. And then, candidate documents are ranked based on their relevance score.
As the amount of information on the web is growing rapidly, search engines have to deal with natural language data on a large scale. The ranking system [13], which plays an important role in search engines, requires an in-depth understanding of the semantics behind queries and documents. Previous work [28, 12, 31] has mainly focused on the design of classification systems by studying the semantic correspondence between query terms and documents. With the advent of pre-trained linguistic models, Eg. BERT [1], existing classification models benefit from learning contextual information from representations of pre-trained terms [25, 20].

The architecture of the proposed TPRM model is shown in Figure 1. This model mainly consists of four modules: (1) User Interest Modeling, (2) UserDoc Interest Matching, (3) QueryDoc Semantic Matching, and (4) Personalized interest rating matching.

For their empirical study, the team compared TPRM with the BM25 algorithm and advanced ad hoc classification models such as KNRM, ConvKNRM, CEDAR KNRM, PClick, SLTB, etc. Experiments were performed on real-world AOL search logs and used mean precision (MAP), mean reciprocal rank (MRR), P@1 (first position precision), and A.Clk (average click position) as metrics to evaluate the quality of the ranking provided. generated.

This model has the following advantages:
The results show that our model can significantly improve TPRM: a topic-based custom ranking model for web search 7 improves the ad hoc ranking model and the custom ranking model. Meanwhile, CEDAR KNRM goes far beyond other ad hoc models, verifying that a pre-trained representation of contextual terms can significantly contribute to the classification system. In addition, most custom ranking models outperform ad hoc ranking models, indicating the effectiveness of user profiles for the ranking system. We also show the performance of TPRMsemantic, i.e. TPRM without the user interest component.

The document provides room for future work:
The ablation studies and further analysis reveal the effects of user interest and semantic correspondence learned from queries and documents.

Results demonstrate that our model can significantly TPRM: A Topic-based Personalized Ranking Model for Web Search 7 improve ad-hoc ranking models and personalized ranking models. Meanwhile, CEDR-KNRM greatly outperforms other ad-hoc models, verifying that pre-trained contextualized term representations can significantly contribute to ranking systems. Moreover, most personalized ranking models outperform ad-hoc ranking models, indicating the effectiveness of user profiles for ranking systems. We also show the performance of TPRM-semantic, which is the TPRM without user interest component. The number of topics is an important hyper-parameter in topic models. To investigate the quality of topics discovered by the topic model, we use the topic coherence score [27] as the evaluation metric.

4. EVALUATION METHODS

To understand the evaluation metrics that can be used for the process of information retrieval and document ranking, we can look at various papers that have evaluated and elucidated the same. The main aim of the paper [3] is to analyze and elucidate various representations of text that are used for retrieving relevant search results, approaches along with the analysis that is carried out in conceptual information retrieval.

Generally, manual interventions are required for evaluating the relationship between the co-related keywords in terms of semantics to replicate the precise results which have paved the way for semantic search. The paper elucidates various representations of text that is used for retrieving relevant search results. Information retrieval started out with searching for keywords in documents. However, this poses a disadvantage as more than one keyword is used for describing and defining a single concept.

The next method that paved its way was a semantic search which involves determining the relationship between related keywords. The paper describes the different ways of representing text that is responsible for retrieving relevant search results. This is done by experimentation and evaluation carried out in conceptual information retrieval.

There have been many methods developed in the past like:

1. HsienTang Lin and Egozi use paragraph based functionality for search system
2. Egozi has developed a MORAG system that demonstrates the BagofWords and ESA methods to perform information retrieval based on concepts
3. Jiang defines macro and micro media metrics to test search results. The macro average is defined as the unweighted average defined across the queries which are referred to as the query-centered measure.
4. Concept-based information retrieval for specific domains proposed by HsienTang Lin
5. Concept-based information retrieval has been developed using various statistical approaches involving Salton vector spatial models.

Many methods have been proposed that address this problem. [23] Real-Time Semantic Search Using Approximate Methodology for Large-Scale Storage Systems. Due to this cloud storage systems mainly fail to offer adequate semantic queries. Exploiting the semantic correlation within and among datasets via correlation-aware hashing and manageable flat-structured addressing to significantly reduce the processing latency, while incurring an acceptably small loss of data-search accuracy can prove to solve the issue [27]. [24] explains Query Expansion Using Local and Global Document Analysis. Analyzing the retrieval of documents by implementing a query and using global analysis techniques, such as word context and phrase structure, on the set of documents and analyzing the results and evaluating their effectiveness. [25] is a Conceptual Representation of Documents and Queries for Information Retrieval Systems by Using Light Ontologies. Analyzing vector space models representing queries based on the concept using WordNet as a light ontology. It also reduces information overlap with respect to classic semantic expansion techniques. [26] explains query expansion using lexical-semantic relations. This paper analyzes the importance of lexical query expansion in the large. Concepts can be presented as WordNet synonym sets and are evaluated by following the typed links included in WordNet. [27] is a personalized query expansion approach for engineering document retrieval. The paper proposes a framework that includes the development of domain ontology, indexing the engineering documents, and performing personalized query expansion and retrieval. User’s interests can be evaluated by querying a relevant document.

The general flow of conceptual-based Information retrieval starts from the web where the extraction of documents happens which are later processed to form a common knowledge base and then both things refer to the indexing of the documents to prepare a ranking. whenever a query is made the data goes through the knowledge base and displays the resultant documents on the basis of the indexes they are ranked.

Conceptual based information retrieval

1. Representation of text
   - 1.1. Concept Vectors
   - 1.2. Semantic Network
2. Approaches
   - 2.1. Statistical based
   - 2.2. Probability-based
   - 2.3. Semantic relatedness
   - 2.4. Genetic Algorithm
3. Sources
   - 3.1. Wikipedia
   - 3.2. Wordnet
   - 3.3. Domain ontology
4. Evaluation

4.1. MAP
4.2. F-measure

Below are the most common evaluation methodologies used:

Precision:
The proportion of documents retrieved that are suited to the user's information need, thus precision measures the exactness of the process. Precision can be calculated by dividing the number of retrieved documents that are relevant by the total number of retrieved documents.

$$Precision = \frac{|\{retrieved\ documents\} \cap \{relevant\ documents\}|}{|\{retrieved\ documents\}|}$$

Recall:
The proportion of the documents that are relevant to the query and were successfully retrieved is known as recall. The recall is calculated by dividing the number of retrieved documents that are relevant by the total number of relevant documents.

$$Recall = \frac{|\{retrieved\ documents\} \cap \{relevant\ documents\}|}{|\{relevant\ documents\}|}$$

Fall-out:
Fall-out is estimated by the fraction of non-relevant documents that are retrieved to all non-relevant documents that are obtainable. Thus, the fall-out is calculated by dividing the number of non-relevant documents that are retrieved by the total number of non-relevant documents present.

$$Fall-out = \frac{|\{retrieved\ documents\} \cap \{non-relevant\ documents\}|}{|\{non-relevant\ documents\}|}$$

F-score:
F-score is the weighted harmonic mean of results obtained from precision and recall metrics.

$$F-score = \frac{2PR}{P+R}\ where\ P\ is\ Precision\ and\ R\ is\ Recall$$

Average precision:
Average precision is used where a sequence of documents is ranked instead of a single document. A precision-recall curve can be plotted using precision as a function of recall. The value of recall (r) ranges from 0 to 1 while p(r) is the value of precision at that point.

$$Average\ precision\ = \int_{0}^{1} p(r)\ dr$$

Mean average precision:
Mean average precision (MAP) is the mean of the average precision values for each query within a set of queries.

$$Mean\ average\ precision = \frac{\sum_{i=1}^{Q} Average\ Precision(i)}{Q}\ where\ Q\ is\ the\ number\ of\ queries$$

ROC curve:
The Receiver Operator Characteristic (ROC) curve plots the Total Positive Rate against the False Positive Rate at different threshold values. Its main purpose is to separate the signal from the noise. The area under this curve measures the ability of the classifier to differentiate between classes. The higher the area under this curve, the better is the performance of the model.

5. DISCUSSION

Bert was released on October 21, 2019, by Google’s development team which was used in Google's search system for English-language queries, including featured snippets. This was one of the biggest improvements in seq-to seq models in the last decade. Following which one of the biggest improvements on BERT was released on Aug 30th, 2020 called DeText or Deep Text Ranking Framework, which marked significant speedup on the processing time in BERT.

TABLE 5.1: The following table differentiates between Bidirectional Encoder Representations from Transformers (BERT) which is a transformer-based machine learning technique Deep Text Ranking Framework (DeText) which is an adaptation of the BERT.

Model	BERT	DeText
Description of the models	BERT is a model that learns contextual embedding. But, this model uses exhaustive iteration over each query word with each document word which is very ineffective. This paper deals with building an efficient BERT-based ranking model.	Representation-based structure: creates query and document embeddings independently instead of applying BERT to a concatenated string of the query and document. This model solution was further extended into a general ranking framework, DeText (Deep Text Ranking Framework)
Datasets/corpora used	MS MARCO Passage Ranking also known as Microsoft Machine Reading Comprehension was used.	Three document ranking datasets were used 1. people search, 2. job search, and 3. help center search.
	It contains 8.8 million unique passages.	5 million queries for people search, 1.5 million queries for job search, 340 thousand queries for help center.

Performance (space and time complexities)/results	A strong interaction-focused seq2seq matching model was made.
	Recall = 81.5%
	Precision = 92.46%
		A similar level of accuracy metric matrix was found for this approach.
		Recall = 83.3%
		Precision = 90.71%

Scope	Using vanilla methods and basic BERT structure we cannot expect exponentially better results.	This model used an exponentially lesser amount of memory space and hence had a faster inference speed, if incorporated with other work in this field we can expect better accuracy 2 papers down the line.

Applications	Considering the size and speed of the model this can’t be used as an enterprise-level model but can be used to develop newer models (BASENET).	Can be used in low compute devices and use cases that don't require high amounts of accuracy but demand faster speeds.
Algorithms/Approaches used	Bidirectional Encoder Representations from Transformers	Distributed Text Embedding Layer and Token Embedding Layer.

From the following comparison, we can conclude that using a distributed text embedding layer with the optimization of the token embedding layer in DeText was responsible for the speedup of BERT and helped make its distributed implementation easier. Seq to Seq models came into the limelight in the early 2014s, which changed the world of Deep Learning as we could now have a variable number of inputs for a given model. COIL of Contextualized Inverted List was released 5 years later and proves to be one of the biggest advancements until this day. It works on a new matching scheme that uses lexical vectors.

TABLE 5.2: Below we have compared one of the most basic Vanilla Seq to Seq models with one of the latest approaches called Contextualized Inverted List or COIL in short.

Model	Vanilla Seq-to-Seq	COIL
Description of the models	This work proposes a novel adaptation of a pre-trained sequence-to-sequence model to the task of the document ranking. This approach is fundamentally different from a commonly-adopted classification-based formulation of ranking, based on encoder-only pre-trained transformer architectures such as BERT.	Contextualized Inverted List works on a new lexical matching scheme that uses vector similarities between query-document. Overlapping term contextualized representations to replace heuristic scoring used in classical systems. It processes documents with deep LM offline and produces representations for each document token.
Datasets/corpora used	MS MARCO Passage Ranking also known as Microsoft Machine Reading Comprehension was used. It contains 8.8 million unique passages.	Two large scale ad hoc retrieval benchmarks from the TREC 2019 Deep Learning (DL) were used: MSMARCO passage (8M English passages of average length around 60 tokens) MSMARCO document (3M English documents of average length around 900 tokens).
Performance (space and time complexities)/results	This model outperforms a classification-based approach, especially in the data-poor regime with limited training data.	COIL-tok achieves an MRR of 0.34 compared to BM25’s MRR 0.18. DeepCT and DocT5Query, which also use deep LMs like BERT and T5.

Scope	Using vanilla methods and basic seq-to-seq structure we cannot expect exponentially better results but can be used as a base for other models.	This approach is more data-efficient than BERT. T5 significantly outperforms BERT when fine-tuned with few training examples. Hence can be used in enterprise-level applications and in use cases that need high accuracy at the cost of higher computational power.
Applications	Considering the size and speed of the model this can't be used as an enterprise-level model but can be used to develop newer models (BASENET).	Enterprise-level applications and use cases where higher accuracy is needed and computational power is in abundance.
Algorithms/Approaches used	Uses a similar masked language modeling objective as BERT to pre-train its encoder-decoder architecture.	Contextualized Exact Lexical Match Scoring and Inverted list indexing.

From the following comparison, we can conclude that using a lexical matching scheme that uses vector similarities between query-documents in COIL was responsible for the high increase in accuracy and the decrease in latency.

SetRank uses a multivariate scoring paradigm while TPRM uses a personalized ranking model as their main algorithm which helps make the embedding layer. Both are improvements of BERT but with different methods to generate embeddings and score them.

TABLE 5.3: Following is the comparison between a multivariate scoring paradigm-based model called SetRank and Topic-based Personalized Ranking Model or TPRM in short.

Model	SetRank	TPRM
Description of the models	SetRank is a new learning-to-rank model based on the multivariate scoring paradigm. The method used is permutation-invariant and can be applied directly to a set of documents with or without any preprocessing. The proposed model in this paper can be considered as a deep model for IR because self-attention networks have been utilized, which is a popular neural technique used in machine learning tasks.	Topic-based Personalized Ranking Model is a model which integrates user topical role with pre-trained contextualized term representations to tailor the general document ranking list. Experiments on the real-world dataset demonstrate that TPRM outperforms state-of-the-art ad-hoc ranking models and personalized ranking models significantly.

Datasets/corpora used	1. Istella LETOR dataset (Istella) 2. Microsoft LETOR 30K (MS LR30K) 3. Yahoo! LETOR challenge set1 (Yahoo)	Dataset constructed by [1] using the real-world AOL search log [23]. Following previous works [1], search logs in the first five weeks are set as history, and the remaining data are used for model training, validation and testing with a ratio of 6:1:1.
Scope	The scoring function in SetRank is designed as a multivariate mapping from a document set to a permutation. Efficiently captures local context information. Naturally involves (multiple) initial rankings High accuracy in ranking.	Results demonstrate that our model can significantly TPRM: A Topic-based Personalized Ranking Model for Web Search 7 improve ad-hoc ranking models and personalized ranking models. Meanwhile, CEDR-KNRM greatly outperforms other ad-hoc models, verifying that pre-trained contextualized term representations can significantly contribute to ranking systems.
Applications	This approach is more data-efficient than BERT. T5 significantly outperforms BERT when fine-tuned with few training examples. Hence can be used in enterprise-level applications and in use cases that need high accuracy at the cost of higher computational power	Most personalized ranking models outperform ad-hoc ranking models, indicating the effectiveness of user-profiles for ranking systems. We also show the performance of TPRM-semantic, which is the TPRM without user interest component.
Algorithms/ Approaches used	Multivariate ranking functions are defined on predefined document sequences. An attention-based neural network module, the Set Transformer, is specifically designed to model interactions among elements in the input set.	Query-Doc Semantic Matching Pretrained language models Mean Average Precision (MAP), Mean Reciprocal Rank (MRR),

From the following comparison, we can conclude that using Query-Doc Semantic Matching Pretrained language models along with Mean Average Precision and Mean Reciprocal Rank in COIL was responsible for the high increase in accuracy and the decrease in latency.

From the rigorous research done across some of the top papers and models used in information retrieval and document ranking the findings we inferred are as follows. Most of the models solved very specific problems and did their job well at the same, each model was trained on huge amounts of data which averaged around 8 million unique paragraphs. If merged together they will be able to make a model/platform which can help encounter most of the problems tackled by each paper and in some cases, some of these are disadvantages of some other papers.

Considering this field in general, the research is extensive, however, there still remains scope for improvement. The possibility of vocabulary mismatch still persists which can be looked into and then worked upon. Additionally, more work could be done on multivariate scoring functions for greater efficiency, even though SetRank has outperformed the traditional learning-to-rank models. Handling a wide range of features and weights still remains a problem.

The efficiency of the BERT is still a major concern as it is on par with all the other metrics except speed. One needs to be careful with tokens with high document frequency, as BERT discards redundant attention weights and periods due to their increased document frequencies. The BERT model can definitely be transformed into a more efficient representation-based model.

6. CONCLUSION

In this work, we have discussed different sections of information retrieval and document ranking and have mentioned different models throughout with their specific perks by which their usage can be seen for different applications. We then presented a thorough and systematic comparison between different models used for the same purpose to help ease the process of finding the right model for an application. Several parts of a retrieval and ranking system have been considered and a final generalized architecture has been suggested that can be useful in a majority of applications.

The architecture proposed divides the entire system into 2 parts: the encoder and decoder. A sequence-to-sequence model has been suggested in association with a loss function and a compression function. The sequence to sequence model is BERT - Bidirectional Encoder Representations from Transformers. In both, the encoder and decoder, the use of multi-head self-attention blocks come into play. There are different layers of BERT units induced which are trained with the Permutation Invariant Ranking as the loss function. Fuzzy hashing can be used as a compression function to differentiate different files by distinguishing their outputs.

In this work, we have discussed different sections of information retrieval and document ranking and have mentioned different models throughout with their specific perks by which their usage can be seen for different applications. We then presented a thorough and systematic comparison between different models used for the same purpose to help ease the process of finding the right model for an application. Several parts of a retrieval and ranking system have been considered and a final generalized architecture has been suggested that can be useful in a majority of applications.

The architecture proposed divides the entire system into 2 parts: the encoder and decoder. A sequence-to-sequence model has been suggested in association with a loss function and a compression function. The sequence to sequence model is BERT - Bidirectional Encoder Representations from Transformers. In both, the encoder and decoder, the use of multi-head self-attention blocks come into play. There are different layers of BERT units induced which are trained with the Permutation Invariant Ranking as the loss function. Fuzzy hashing can be used as a compression function to differentiate different files by distinguishing their outputs.

The entire model can be rolled out across time to better visualize the process of taking sequences as inputs and representing the model in a Recurrent manner. The first part, the Encoder, is the module that takes the document(s) as its input and stores the targeted information in its weights. The following models are present in the encoder: Multi-Head Attention block: Multi-threaded attention blocks help in capturing a higher amount of feature vectors that will be considered to calculate inference. The attention blocks are followed by Transformer units. These are the blocks that comprise 95% of the model. Between every unit, there is an Add & Normalization layer which adds the output of the last sequence iteration and the current logits.

The second part of the model is the Decoder which takes two inputs namely the logits/weights from the encoder and the input query. The logits/weights are responsible for saving user data from the input documents and the query is the question asked. It comprises the following modules: Multi-Head Attention block: Multi-threaded attention blocks in this case help in capturing a higher amount of feature vectors from the input query which will be inferred against the logits created by the encoder. The attention blocks are followed by Transformer units, which in this case have word embeddings and output. Between every unit, there is an Add & Normalization layer that adds the output of the last sequence iteration and the current logits. Finally, each output of the decoders is summed together in a basic dense neural network that uses tan-has the output activation function.

7. FUTURE WORK

Based on the analysis of different models, the problem of all-to-all match retrievers from documents is still a prevalent issue in most of them. This process is time-consuming and many researchers have proposed models like DeText [1] to improve it, but none of the suggested models is known widely. Improvement in this field remains an open journal for new research. The way a model exploits the knowledge given to it and trains itself according to the knowledge to generate fluent text is still rudimentary. That exploitability is a key to developing high-quality texts and explanations based on the query proposed by the users.

Further, there could be more work done on multivariate scoring functions for greater efficiency, even though SetRank has outperformed the traditional learning-to-rank models. In addition to this, datasets used can be improved and a past model can be run on it to check its efficiency on the basis of which parameters that are lagging can be worked upon. This provides the most opportunity for researchers seeking to make significant improvements in the field of information retrieval and document ranking.

DECLARATIONS

The authors have no relevant financial or non-financial interests to disclose. Also, the authors have no proprietary interests in any material discussed in this article.

REFERENCES

1. Guo, W., Liu, X., Wang, S., Gao, H., Sankar, A., Yang, Z., ... & Agarwal, D. (2020, October). Detext: A deep text ranking framework with bert.
2. Gao, L., Dai, Z., & Callan, J. (2021). COIL: Revisit Exact Lexical Match in Information Retrieval with Contextualized Inverted List. arXiv preprint arXiv:2104.07186.
3. Mahalakshmi, P., & Fatima, N. S. (2021). An Art of Review on Conceptual-based Information Retrieval. Webology, 18(1), 51-61.
4. Pang, L., Xu, J., Ai, Q., Lan, Y., Cheng, X., & Wen, J. (2020, July). Setrank: Learning a permutation-invariant ranking model for information retrieval.
5. Zhan, J., Mao, J., Liu, Y., Zhang, M., & Ma, S. (2020, July). An analysis of BERT in document ranking.
6. Nogueira, R., Jiang, Z., & Lin, J. (2020). Document ranking with a pretrained sequence-to-sequence model. arXiv preprint arXiv:2003.06713.
7. Ostendorff, Malte & Ruas, Terry & Blume, Till & Gipp, Bela & Rehm, Georg. (2020). Aspect-based Document Similarity for Research Papers.
8. Huang, Minghui & Peng, Wei & Wang, Dong. (2021). TPRM: A Topic-based Personalized Ranking Model for Web Search.
9. Xiang, Biao & Jiang, Daxin & Pei, Jian & Sun, Xiaohui & Chen, Enhong & Li, Hang. (2010). Context-aware ranking in web search. 451-458. 10.1145/1835449.1835525.
10. Hofstätter, S., Zamani, H., Mitra, B., Craswell, N., & Hanbury, A. (2020, July). Local self-attention over long text for efficient document retrieval.
11. Suma, V. (2020). A novel information retrieval system for distributed cloud using hybrid deep fuzzy hashing algorithm. JITDW, 2(03), 151-160.
12. Yang, Y., Qiao, Y., Shao, J., Anand, M., Yan, X., & Yang, T. (2021). Composite Re-Ranking for Efficient Document Search with BERT. arXiv preprint arXiv:2103.06499.
13. Guo, J., Fan, Y., Ai, Q., & Croft, W. B. (2016, October). A deep relevance matching model for ad-hoc retrieval. In Proceedings of the 25th ACM international conference on information and knowledge management (pp. 55-64).
14. Palangi, H., Deng, L., Shen, Y., Gao, J., He, X., Chen, J., ... & Ward, R. (2016). 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), 694-707.
15. Shen, Y., He, X., Gao, J., Deng, L., & Mesnil, G. (2014, November). A latent semantic model with convolutional-pooling structure for information retrieval. In Proceedings of the 23rd ACM international conference on conference on information and knowledge management (pp. 101-110).

16. Zamani, H., Mitra, B., Song, X., Craswell, N., & Tiwary, S. (2018, February). Neural ranking models with multiple document fields. In Proceedings of the eleventh ACM international conference on web search and data mining (pp. 700-708).
17. Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural computation, 9(8), 1735-1780.
18. Yi Luan, Jacob Eisenstein, Kristina Toutanova, and M. Collins. 2020. Sparse, dense, and attentional representations for text retrieval. ArXiv, abs/2005.00181.
19. R. Guo, Philip Y. Sun, E. Lindgren, Quan Geng, David Simcha, Felix Chern, and S. Kumar. 2019. Accelerating large-scale inference with anisotropic vector quantization. arXiv: Learning
20. Zhuyun Dai and J. Callan. 2019a. Context-aware sentence/passage term importance estimation for first stage retrieval. ArXiv, abs/1910.10687
21. Tomas Mikolov, Ilya Sutskever, Kai Chen, G. S. Corrado, and J. Dean. 2013. Distributed representations of words and phrases and their compositionality. In NIPS.
22. Z. Yang, Zihang Dai, Yiming Yang, J. Carbonell, R. Salakhutdinov, and Quoc V. Le. 2019. Xlnet: Generalized autoregressive pre-training for language understanding. In NeurIPS
23. Hua, Y., Jiang, H., & Feng, D. (2015). Real-time semantic search using the approximate methodology for large-scale storage systems. IEEE Transactions on Parallel and Distributed Systems, 27(4), 1212-1225.
24. Jinxi, X., & Bruce Croft, W. (1996). Query Expansion Using Local and Global Document Analysis. Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval, 4-11.
25. Dragoni, M., Da Costa Pereira, C., & Tettamanzi, A.G. (2012). A conceptual representation of documents and queries for information retrieval systems by using light ontologies. Expert Systems with applications, 39(12), 10376-10388.
26. Voorhees, E.M. (1994). Query expansion using lexical-semantic relations. In SIGIR'94, Springer, London, 61-69.
27. Hahm, G.J., Yi, M.Y., Lee, J.H., & Suh, H.W. (2014). A personalized query expansion approach for engineering document retrieval. Advanced Engineering Informatics, 28(4), 344-359.
28. Ai, Q., Bi, K., Guo, J., & Croft, W. B. (2018, June). Learning a deep listwise context model for ranking refinement. In The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval (pp. 135-144).
29. Bello, I., Kulkarni, S., Jain, S., Boutilier, C., Chi, E., Eban, E., ... & Meshi, O. (2018). Seq2slate: Re-ranking and slate optimization with rnns. arXiv preprint arXiv:1810.02019.
30. Jiang, R., Gowal, S., Mann, T. A., & Rezende, D. J. (2018). Beyond greedy ranking: Slate optimization via list-CVAE. arXiv preprint arXiv:1803.01682.
31. Pasumarthi, R. K., Wang, X., Bendersky, M., & Najork, M. (2019). Self-attentive document interaction networks for permutation equivariant ranking. arXiv preprint arXiv:1910.09676.
32. Pei, C., Zhang, Y., Zhang, Y., Sun, F., Lin, X., Sun, H., ... & Pei, D. (2019, September). Personalized re-ranking for a recommendation. In Proceedings of the 13th ACM Conference on Recommender Systems (pp. 3-11).
33. Dai, Z., & Callan, J. (2019, July). Deeper text understanding for IR with contextual neural language modeling. In Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval (pp. 985-988).
34. Nogueira, R., & Cho, K. (2019). Passage Re-ranking with BERT. arXiv preprint arXiv:1901.04085.
35. Clark, K., Khandelwal, U., Levy, O., & Manning, C. D. (2019). What does bert look at? an analysis of bert's attention. arXiv preprint arXiv:1906.04341.
36. Ethayarajh, K. (2019). How contextual are contextualized word representations? comparing the geometry of BERT, ELMo, and GPT-2 embeddings. arXiv preprint arXiv:1909.00512.
37. Qiao, Y., Xiong, C., Liu, Z., & Liu, Z. (2019). Understanding the Behaviors of BERT in Ranking. arXiv preprint arXiv:1904.07531.
38. Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.
39. S. E. Robertson, S. Walker, S. Jones, M. Hancock-Beaulieu, and M. Gatford. Okapi at TREC-3. In Proceedings of the 3rd Text Retrieval Conference (TREC-3), pages 109–126, 1994.
40. Kudo, T., & Richardson, J. (2018). Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. arXiv preprint arXiv:1808.06226.
41. Asadi, N., & Lin, J. (2013, July). Effectiveness/efficiency tradeoffs for candidate generation in multi-stage retrieval architectures. In Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval (pp. 997-1000).
42. Nogueira, R., Yang, W., Cho, K., & Lin, J. (2019). Multi-stage document ranking with bert. arXiv preprint arXiv:1910.14424.

43. Bajaj, P., Campos, D., Craswell, N., Deng, L., Gao, J., Liu, X., ... & Wang, T. (2016). Ms Marco: A human-generated machine reading comprehension dataset. arXiv preprint arXiv:1611.09268.
44. E. M. Voorhees. Overview of the TREC 2004 Robust Track. In Proceedings of the Thirteenth Text Retrieval Conference (TREC 2004), pages 52–69, 2004.
45. Daniel Bar, Torsten Zesch, and Iryna Gurevych. 2011. A Reflective View on Text Similarity. ¨ International Conference Recent Advances in Natural Language Processing (RANLP), pages 515–520.
46. Joel Chan, Joseph Chee Chang, Tom Hope, Dafna Shahaf, and Aniket Kittur. 2018. SOLVENT: A Mixed-Initiative System for Finding Analogies between Research Papers. Proceedings of the ACM on Human-Computer Interaction, 2(CSCW):1–21, nov.
47. Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention Is All You Need. Proceedings of the 31st International Conference on Neural Information Processing Systems, pages 6000–6010, Jun.
48. Arman Cohan, Sergey Feldman, Iz Beltagy, Doug Downey, and Daniel S. Weld. 2020. SPECTER: Document-level Representation Learning using Citation-informed Transformers. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL).
49. Malte Ostendorff, Peter Bourgonje, Maria Berger, Julian Moreno-Schneider, Georg Rehm, and Bela Gipp. 2019. Enriching BERT with Knowledge Graph Embeddings for Document ranking. In Proceedings of the 15th Conference on Natural Language Processing (KONVENS 2019), pages 305–312, Erlangen, Germany. German Society for Computational Linguistics & Language Technology.
50. Ahmad, W.U., Chang, K., Wang, H.: Context attentive document ranking and query suggestion. In: SIGIR. pp. 385–394. ACM (2019)
51. Bennett, P.N., White, R.W., Chu, W., Dumais, S.T., Bailey, P., Borisyuk, F., Cui, X.: Modeling the impact of short- and long-term behavior on search personalization. In: SIGIR. pp. 185–194. ACM (2012)
52. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent Dirichlet allocation. In: NIPS. pp. 601– 608. MIT Press (2001)
53. Carman, M.J., Crestani, F., Harvey, M., Baillie, M.: Towards query log based personalization using topic models. In: CIKM. pp. 1849–1852. ACM (2010)
54. Chirita, P., Nejdl, W., Paiu, R., Kohlsch¨utter, C.: Using ODP metadata to personalize search. In: SIGIR. pp. 178–185. ACM (2005)
55. Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. SimCSE: Simple Contrastive Learning of Sentence Embeddings. CoRR abs/2104.08821 (2021). arXiv:2104.08821 https://arxiv.org/abs/2104.08821
56. Andrew M. Dai and Quoc V. Le. 2015. Semi-supervised Sequence Learning. In Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada, Corinna Cortes, Neil D. Lawrence, Daniel D. Lee, Masashi Sugiyama, and Roman Garnett (Eds.). 3079–3087.https://proceedings.neurips.cc/paper/2015/hash/7137debd45ae4d0ab9aa953017286b20-Abstract.html
57. Baotian Hu, Zhengdong Lu, Hang Li, and Qingcai Chen. 2014. Convolutional Neural Network Architectures for Matching Natural Language Sentences. In Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada, Zoubin Ghahramani, Max Welling, Corinna Cortes, Neil D. Lawrence, and Kilian Q. Weinberger (Eds.). 2042–2050. https://proceedings.neurips.cc/ paper/2014/hash/b9d487a30398d42ecff55c228ed5652b-Abstract.html
58. Songwei Ge, Zhicheng Dou, Zhengbao Jiang, Jian-Yun Nie, and Ji-Rong Wen. 2018. Personalizing Search Results Using Hierarchical RNN with Query-aware Attention. In Proceedings of the 27th ACM International Conference on Information and Knowledge Management, CIKM 2018, Torino, Italy, October 22-26, 2018, Alfredo Cuzzocrea, James Allan, Norman W. Paton, Divesh Srivastava, Rakesh Agrawal, Andrei Z. Broder, Mohammed J. Zaki, K. Selçuk Candan, Alexandros Labrinidis, Assaf Schuster, and Haixun Wang (Eds.). ACM, 347–356. https://doi.org/10.1145/ 3269206.3271728
59. Xiong, C., Dai, Z., Callan, J., Liu, Z., & Power, R. (2017, August). End-to-end neural ad-hoc ranking with kernel pooling. In Proceedings of the 40th International ACM SIGIR conference on research and development in information retrieval (pp. 55-64).
60. Zamani, H., Dehghani, M., Croft, W. B., Learned-Miller, E., & Kamps, J. (2018, October). From neural re-ranking to neural ranking: Learning a sparse representation for inverted indexing. In Proceedings of the 27th ACM international conference on information and knowledge management (pp. 497-506).
61. MacAvaney, S., Yates, A., Cohan, A., & Goharian, N. (2019, July). CEDR: Contextualized embeddings for document ranking. In Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval (pp. 1101-1104).

62. Yan, M., Li, C., Wu, C., Bi, B., Wang, W., Xia, J., & Si, L. (2019). IDST at TREC 2019 Deep Learning Track: Deep Cascade Ranking with Generation-based Document Expansion and Pre-trained Language Modeling. In TREC.
63. Kitaev, N., Kaiser, Ł., & Levskaya, A. (2020). Reformer: The efficient transformer. arXiv preprint arXiv:2001.04451.
64. Craswell, N., Mitra, B., Yilmaz, E., Campos, D., & Voorhees, E. M. (2020). Overview of the trec 2019 deep learning track. arXiv preprint arXiv:2003.07820.
65. Zhu, Y., Yan, E., & Song, I. Y. (2017). A natural language interface to a graph-based bibliographic information retrieval system. Data & Knowledge Engineering, 111, 73-89
66. Farhi, S. H., & Boughaci, D. (2018). Graph-based model for information retrieval using a stochastic local search. Pattern Recognition Letters, 105, 234-239.
67. Kulunchakov, A. S., & Strijov, V. V. (2017). Generation of simple structured information retrieval functions by a genetic algorithm without stagnation. Expert Systems with Applications, 85, 221-230.
68. Lai, J., Mu, Y., Guo, F., Jiang, P., & Susilo, W. (2018). Privacy-enhanced attribute-based private information retrieval. Information sciences, 454, 275-291.
69. Raj, J. S. (2019). Efficient information maintenance using computational intelligence in the multi-cloud architecture. Journal of Soft Computing Paradigm (JSCP), 1(02), 113-124
70. Caruana, R., Joachims, T., & Backstrom, L. (2004). KDD-Cup 2004: results and analysis. ACM SIGKDD Explorations Newsletter, 6(2), 95-108.
71. MacAvaney, S., Yates, A., Cohan, A., & Goharian, N. (2019, July). CEDR: Contextualized embeddings for document ranking. In Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval (pp. 1101-1104).
72. Mitra, B., Hofstatter, S., Zamani, H., & Craswell, N. (2020). Conformer-kernel with query term independence for document retrieval. arXiv preprint arXiv:2007.10434.
73. Khattab, O., & Zaharia, M. (2020, July). Colbert: Efficient and effective passage search via contextualized late interaction over bert. In Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval (pp. 39-48).
74. Shao, J., Ji, S., & Yang, T. (2019, July). Privacy-aware document ranking with neural signals. In Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval (pp. 305-314).
75. Zhuang, H., Wang, X., Bendersky, M., Grushetsky, A., Wu, Y., Mitrichev, P., ... & Qian, H. (2020). Interpretable Learning-to-Rank with Generalized Additive Models. arXiv preprint arXiv:2005.02553.