



Utilizing BERT for Information Retrieval: Survey, Applications, Resources, and Challenges

JIAJIA WANG, School of Sciences, Henan University of Technology, Zhengzhou, China

JIMMY XIANGJI HUANG, Information Retrieval and Knowledge Management Research Lab, York University, Toronto, Canada

XINHUI TU, School of Computer Science, Central China Normal University, Wuhan, China

JUNMEI WANG, School of Computer, Hangzhou Dianzi University, Hangzhou, China

ANGELA JENNIFER HUANG, Lassonde School of Engineering, York University, Toronto, Canada

MD TAHMID RAHMAN LASKAR, York University & Dialpad Inc., Toronto, Canada

AMRAN BHUIYAN, Information Retrieval and Knowledge Management Research Lab, York University, Toronto, Canada

Recent years have witnessed a substantial increase in the use of deep learning to solve various natural language processing (NLP) problems. Early deep learning models were constrained by their sequential or unidirectional nature, such that they struggled to capture the contextual relationships across text inputs. The introduction of bidirectional encoder representations from transformers (BERT) leads to a robust encoder for the transformer model that can understand the broader context and deliver state-of-the-art performance across various NLP tasks. This has inspired researchers and practitioners to apply BERT to practical problems, such as information retrieval (IR). A survey that focuses on a comprehensive analysis of prevalent approaches that apply pretrained transformer encoders like BERT to IR can thus be useful for academia and the industry. In light of this, we revisit a variety of BERT-based methods in this survey, cover a wide range of techniques of IR, and group them into six high-level categories: (i) handling long documents, (ii) integrating semantic information, (iii) balancing effectiveness and efficiency, (iv) predicting the weights of terms, (v) query expansion, and (vi) document expansion. We also provide links to resources, including datasets and toolkits, for BERT-based IR systems. Additionally, we highlight the advantages of employing encoder-based BERT models in contrast to recent large language models like ChatGPT, which are decoder-based and demand extensive computational resources. Finally, we summarize the comprehensive outcomes of the survey and suggest directions for future research in the area.

CCS Concepts: • **Information systems** → **Information retrieval;**

This research is supported by the research grant (RGPIN-2020-07157) from the Natural Science and Engineering Research Council (NSERC) of Canada, the York Research Chairs (YRC) program, and an Ontario Research Fund-Research Excellence (ORF-RE) award from the BRAIN Alliance.

Authors' addresses: J. Wang, School of Sciences, Henan University of Technology, Zhengzhou, Henan 450001, China; e-mail: stwj057@mails.cenu.edu.cn; J. X. Huang (Corresponding author) and A. Bhuiyan, Information Retrieval and Knowledge Management Research Lab, York University, 4700 Keele St., Toronto, Ontario M3J 1P3, Canada; e-mail: jhuang@yorku.ca & amran@yorku.ca; X. Tu, School of Computer Science, Central China Normal University, Wuhan, Hubei 430079, China; e-mail: tuxinhui@mail.cenu.edu.cn; J. Wang, School of Computer, Hangzhou Dianzi University, Hangzhou 310018, China; e-mail: wjm2018@mails.cenu.edu.cn; A. J. Huang, Lassonde School of Engineering, York University, Toronto, Ontario M3J 2S5, Canada; e-mail: anqiaj@yorku.ca; M. T. R. Laskar, York University & Dialpad Inc., 4700 Keele St. Toronto, Ontario M3J 1P3, Canada; e-mail: tahmid20@yorku.ca.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2024 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM 0360-0300/2024/04-ART185

<https://doi.org/10.1145/3648471>

Additional Key Words and Phrases: BERT, information retrieval, natural language processing, artificial intelligence

ACM Reference Format:

Jiajia Wang, Jimmy Xiangji Huang, Xinhui Tu, Junmei Wang, Angela Jennifer Huang, Md Tahmid Rahman Laskar, and Amran Bhuiyan. 2024. Utilizing BERT for Information Retrieval: Survey, Applications, Resources, and Challenges. *ACM Comput. Surv.* 56, 7, Article 185 (April 2024), 33 pages. <https://doi.org/10.1145/3648471>

1 INTRODUCTION AND MOTIVATION

Information retrieval (IR) is the acquisition of informational resources from a large volume of unstructured information that satisfies the user's needs [37]. Web search engines are a typical example of an information retrieval system. IR models are used to sort and rank the retrieved documents according to the user's queries; this is essential for designing search engines. Traditional IR models include the *vector space model*, the *probability model*, and the *language model*. In 1959, Arthur Samuel introduced the term *machine learning* as a method of data analysis that automates the building of an analytical model. Machine learning is a branch of **Artificial Intelligence (AI)** based on the idea that the system can learn from data, identify patterns, and make decisions with minimal human intervention.

Learning to rank (LTR) is an approach that involves utilizing machine learning techniques to construct ranking models. Unlike traditional and manually constructed ranking formulae, LTR uses data to automatically build a model that can make predictions without being explicitly programmed to do so. While it delivers superior performance to traditional IR models, its capabilities can be further enhanced in two areas. First, the complexity and diversity of languages and cultures make it challenging to accurately comprehend semantic information embedded within queries and document content. Second, feature engineering is manually performed in machine learning and requires input from domain experts, where this incurs a significant cost.

Deep learning is a domain of machine learning that uses artificial neural networks with multiple layers to learn patterns from the given data. It has significantly outperformed traditional machine learning models in recent years to become the focus of AI research. Deep learning has exhibited an impressive capability in a wide range of tasks, such as speech recognition [32], natural language processing [28], recommender systems [118], and computer vision [8]. For more information on deep neural networks, we refer the interested reader to Reference [126] and Reference [7].

In early research in the area, vectors of word embedding like word2vec¹ [121] and GloVe² [51] were used in neural network models to obtain the representation of the words in the input texts. Word2vec uses one of the following two models to this end: the (i) **continuous bag-of-words (CBOW)** model or the (ii) continuous skip-gram model. The CBOW model predicts a given word according to the words surrounding it within a fixed window, where the order of the surrounding words does not impact the results of its predictions. By contrast, a continuous skip-gram predicts the surrounding words in a fixed window based on the given word. This causes the nearby contextual words to have a more significant influence on the predicted word than the contextual words that are more distant. **Global Vectors (GloVe)** is a tool for word representation based on global term frequency. The word representations generated by GloVe capture some semantic information between words. However, neither word2vec nor GloVe can adequately capture contextual information. Contextualized information refers to the intelligence within the given discourse that exploits

¹<https://radimrehurek.com/gensim/models/word2vec.html>

²<https://nlp.stanford.edu/projects/glove/>

the relationships between the words, phrases, sentences, and even paragraphs by considering all the circumstances (e.g., bidirectional contexts) involved in the emergence of the text. For example, the word *bank* appears in *bank account* and *bank of the river*. If a model generates the same representation for *bank* in both scenarios, then this means that it does not capture the contextual information. A context-based model can generate different representations for *bank* based on the words preceding and following it (i.e., bidirectional contextual information) in the sentence.

Considering the importance of bidirectional contextual information, Devlin et al. proposed **Bidirectional Encoder Representations from Transformers (BERT)** [44] in 2018. It leverages the transformer encoder [12] to pretrain a bidirectional language model. BERT has demonstrated an impressive capability in terms of understanding language in various **natural language processing (NLP)** tasks: (i) It outperformed previously proposed language models [10, 73] on the **General Language Understanding Evaluation (GLUE)** [13] benchmark and delivered state-of-the-art performance on tasks like natural language inference, textual entailment, sentiment analysis, and sentence similarity modeling. (ii) It demonstrated impressive comprehension on the **Stanford Question Answering Dataset (SQuAD)** v1.1 test set by outperforming even humans on two measures of evaluation (ExactMatch and the F1 score). (iii) BERT has a good capacity for transfer learning [90, 144] that helps it adapt to different tasks through pretraining or fine-tuning.

The successful use of BERT in various applications involving language comprehension has also inspired researchers to apply BERT-based approaches to IR [9, 78, 81, 107, 120]. In light of the recent successes of BERT-based models in IR, and given the significant number of studies in this domain, we provide a survey in this study that brings together a number of BERT-based approaches into a coherent resource that provides a single point reference for research on IR. Note that few surveys [21, 47, 61] in the literature have highlighted neural approaches to IR. For instance, Onal et al. [61] focused on the fundamentals of text information retrieval, while Mitra et al. [21] presented a brief review of IR methods that use pretrained embeddings and neural networks. Closer to our work here, Guo et al. [47] surveyed IR approaches that focus only on ranking models using deep neural networks. Unlike these approaches, our survey targets IR approaches that utilize pretrained transformer encoders like BERT and its variants while also systematically reviewing the challenges of using BERT-based models for IR.

Since 2022, there has been remarkable progress in leveraging **Language Models (LMs)**, encompassing **Pretrained Language Models (PLMs)** and **Large Language Models (LLMs)**. PLMs such as BERT [44], RoBERTa [140], and **generative pretrained transformer (GPT)** [10] undergo pretraining on extensive textual corpora to assimilate worldly knowledge. The recent advent of LLMs, exemplified by OpenAI's ChatGPT [80] and GPT-4 [87], has catalyzed an unforeseen surge in innovation, highlighting the diverse possibilities of AI.

While decoder-only LLMs like ChatGPT³ [80] have gained significant attention for their ability to generate humanlike text, there remain many challenges in using LLMs in real-world problems. For instance, they require significant computational resources for both training and inference. In addition, they can generate responses based only on their pretraining knowledge but cannot answer any time-sensitive questions that are beyond their pretraining timeline. Moreover, using LLMs as a service by utilizing their APIs poses privacy-related concerns. Considering their cost, failure to adapt over time, and privacy-related issues, it is challenging to use such LLMs in real-world applications. As an alternative, encoder-based transformer models like BERT do not pose privacy-related risks and require significantly lower computational resources. BERT-based models are particularly adept at information retrieval tasks [30] that are crucial for applications like search engines and recommendation systems owing to their effective semantic understanding, low

³<https://chat.openai.com/>

computational requirements, and efficient response times. BERT-based rich contextual embeddings are useful for retrieving information from external resources to augment the context of the LLM during inference [88]. In this regard, we aim to review the use of pretrained transformer encoder models like BERT in IR. This can help fill a gap in the relevant research and can provide a comprehensive overview for researchers and industry professionals.

The major contributions of this survey in this context are fourfold. First, we logically organize the aspects of BERT for ad hoc IR in terms of the updated literature. This enables the reader to understand BERT-related models from various perspectives in a step-by-step manner and discover new research opportunities by following this organization. Second, we analyze the advantages and disadvantages of different models based on BERT, and conduct a number of state-of-the-art comparisons in different categories. Third, we present the challenges to the use of BERT for IR and examine important IR toolkits. Finally, we provide comprehensive conclusions based on the outcomes of the survey and outline the trend of future research on BERT-based IR.

The remainder of this article is organized as follows: We discuss the background and structure of BERT in Section 2. Section 3 discusses the use of BERT for ad hoc IR, including utilizing BERT for handling long documents, integrating semantic information, balancing effectiveness and efficiency, BERT-based term weight prediction, BERT-based query expansion, and BERT-based document expansion, respectively. In Section 4, we present resources that incorporate datasets as well as BERT-based ranking models. Finally, we provide the conclusions of this survey in Section 5.

2 BACKGROUND

The popularization of the Web and the explosive growth of Internet-based information have led researchers to focus on the performance and efficiency of IR systems as well as the technology of their core search engines. In addition to crawling and indexing in the pipeline, the most important factor while designing a search engine is ranking, which relies solely on IR models for sorting the relevant documents with respect to a user's query. Thus, the efficiency of a search engine heavily relies on the efficiency of the IR model.

Relevance-based ranking is the main problem in IR that plays a fundamental role in various applications of it, such as search engines. To satisfy the user's informational needs (queries), we need to identify the collection of documents that is the most relevant to their queries. Given a set of queries $Q = \{q_1, q_2, \dots, q_m\}$ and a set of documents $D = \{d_1, d_2, \dots, d_n\}$, we need to use an IR model to rank the relevant documents based on search queries.

In this section, we discuss the most popular traditional IR models in Section 2.1, neural ranking models in Section 2.2, pretrained language models in Section 2.3, and improvements and extensions of pretrained language models in Section 2.4.

2.1 Traditional Retrieval Models

Traditional retrieval models can be divided into four categories: Boolean models, vector space models, language models, and probability models. The first three models assume that documents and queries are composed of a set of words and ignore the positional information of tokens in a given query/document.

The Boolean model [6, 34] is a simple retrieval model based on set theory and Boolean algebra. It aims to find documents that return “1” with a query term. However, it lacks the concept of document rank, which limits its retrieval capabilities. Salton et al. introduced a non-binary weight model called the **vector space model (VSM)** [33] to rank the relevant documents based on search queries. The VSM [33] transforms text processing into vector computation in vector space, presents semantic similarity through spatial similarity, and thus compensates for the Boolean model's inability to scale the returned documents. However, approaches based on the VSM model fail to

recognize semantic ambiguity in the text due to their dependence on token frequency and cannot solve the problems of polysemy and synonymy. The **Best Matching 25 (BM25)** probabilistic model [114] is among the best weighting models for IR [16, 27, 54–56, 59, 60, 94, 134, 139]. It assigns a weight to a search term based on its intra-document term frequency and query term frequency. The document weight is the sum of the weights of the query term. Ponte et al. [53] proposed a language model that considers a passage of natural language text as a discrete time series. They utilized probabilistic models to estimate the probability of a document being relevant based directly on a particular probability distribution and a language model to predict the probability of a given sequence of words occurring in a sentence.

2.2 Neural Ranking Models

Traditional retrieval models obtain ranking scores by searching for exact matches of words in both the query and the document. However, this limits the availability of positional and semantic information and may lead to *vocabulary mismatch* [125]. By contrast, neural ranking models construct query-to-document relevance structures to learn feature representations automatically via word2vec [121] and GloVe [51]. There are two main classes of neural ranking models: representation-focused models and interaction-focused models.

- *Representation-focused Architecture*: This architecture encodes the query and the document separately. Thus, no interaction occurs between the query and the document during the encoding procedure. Following this pipeline, Huang et al. introduced the **deep structured semantic model (DSSM)** [89] that can predict the semantic similarity between sentences and obtain a lower-dimensional semantic vector representation of a sentence. Similarly, Hu et al. in 2014 studied two models for text matching [17], **Architecture I (ARC-I)** and ARC-II. The ARC-I model uses a **convolutional neural network (CNN)** to separately obtain features from texts and calculate the similarities between the features. However, there is no interaction between the texts while the features are extracted.

- *Interaction-focused Architecture*: This architecture evaluates the relationship between a query and a document according to their interaction. Compared with representation-focused architecture, interaction-focused architecture can capture more contextualized information. Hu et al. proposed the ARC-II model [17], which uses a one-dimensional (1D) convolutional layer to explore the interaction between input texts. The **Deep Relevance Matching Model (DRMM)** [46], introduced by Guo et al., uses a histogram to map the relevance between the query and words in the document. Wan et al. proposed **Match Spatial Recurrent Neural Networks (SRNN)** [116], which models complex interactions between input texts by using a neural tensor layer. Xiong et al. proposed **Kernelized Neural Ranking Model (KNRM)** [18], which uses a kernel pooling layer to calculate the relevance between the query and words in the document. Dai et al. introduced Conv-KNRM [149], a model for the soft matching of n -grams for ad hoc IR by using a CNN. Although all these approaches improve performance over a range of IR applications, they fail to consistently work well when trained on small-scale data.

2.3 Pretrained Language Models

Creating a better language model is crucial to NLP. In the broad sense, a pretrained language model refers to a language model trained on large-scale data, including distributed static representation-based models (word2vec [121] and GloVe [51]) and dynamic contextualized models (CoVe [19] and **Embeddings from Language Models (ELMo)** [73]). Following the new wave of deep transformer-focused pretrained models [10], such as the GPT [10] developed by OpenAI, a new understanding of the pretrained language model, known as BERT [44], emerged in the NLP community in 2018. Compared with previous models [17, 18, 89, 149], pretrained models based on

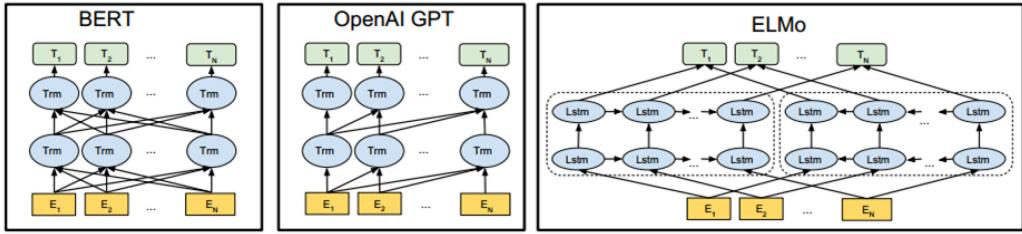


Fig. 1. The pretraining model architectures of BERT, OpenAI GPT, and ELMo. BERT demonstrates a significant bidirectional nature, OpenAI’s GPT exhibits a unidirectional characteristic, while ELMo uses a shallow bidirectional approach.

deep transformer encoders have the advantage of being able to process input text sequences in parallel. In addition, transformer encoder-based pretrained language models can focus on the target task and fine-tune it to achieve better performance than the previously mentioned models [17, 18, 89, 149], where the representation of word vectors remains constant when used as an additional input to perform the target task.

A comparison of pretrained transformer models reveals that the GPT model from OpenAI uses a multi-layer decoder based on a transformer. It is mainly composed of two parts: generative pre-training and differential fine-tuning. Unlike the GPT, BERT does not leverage the decoder of the transformer for self-regression language modeling. Instead, it introduces the auto-encoding pre-training task. In contrast to ELMo, which uses the left-to-right and right-to-left **Long Short-Term Memory (LSTM)** models, BERT uses the bidirectional transformer encoder as the language model and takes the **masked language model (MLM)** and **next sentence prediction (NSP)** as the objectives of pretraining. Figure 1 shows a comparison of ELMo, OpenAI GPT, and BERT.

2.4 Improvements to and Extensions of Pretrained Language Models

The advantage of BERT’s auto-encoding language model is that it can use contextualized information on the given word to better predict its vector representation. Because BERT has pioneered a new area of NLP, many variants and extensions based on it have emerged.

Yang et al. introduced XLNET [154] in 2019. It is an auto-regressive language model based on Transformer-XL [145], in which all tokens are predicted in random order, rather than sequentially, and the joint probability of each predicted token is factorized by using the product rule to address the issue of a discrepancy between pretraining and fine-tuning in BERT due to the use of artificial symbols like [MASK]. In addition, XLNET incorporates the advantages of the auto-encoding language model and obtains contextualized information by introducing the permutation language model and two-stream self-attention. Shortly after XLNET overtook BERT across the board, Facebook proposed the **Robustly Optimized BERT Pretraining Approach (RoBERTa)** [140], which can achieve state-of-the-art performance on multiple tasks. RoBERTa does not change the structure of Google’s BERT but only its method of pretraining. The changes in the pretraining methods are mainly reflected in the following aspects: (1) Static masking vs. dynamic masking: In static masking, the masking operation is performed only once during data processing, while in dynamic masking, the pattern of masking changes whenever a sequence is fed into the model; (2) with NSP vs. without NSP (removing the NSP from RoBERTa’s training procedure enables the input to be packed with full sentences sampled from different documents until the maximum length of the sequence is 512; this is called FULL-SENTENCES); (3) a larger mini-batch; and (4) more training data and a longer training time.

As with BERT, XLNET and RoBERTa encounter the issue of consistent growth in the size of the pretrained language model that leads to memory-related constraints, a longer preparation time, and unexpectedly poor performance. To reduce memory consumption and improve the speed of training of the model, Lan et al. [150] proposed **A Lite BERT (ALBERT)**, which incorporates two parameter-reduction techniques: word vector decomposition and cross-layer parameter sharing. Pretraining in XLNET and RoBERTa by using NSP is simple and thus cannot be used to learn deep semantic information. ALBERT replaces it with **sentence-order prediction (SOP)** to predict if the sentences are coherent. Different from the above-mentioned pretrained models, Clark et al. proposed **Efficiently Learning an Encoder that Classifies Token Replacements Accurately (ELECTRA)** [62], which combines BERT with a structure similar to that of the generative adversarial network [42]. ELECTRA consists of a generator and a discriminator and can outperform BERT while using fewer parameters and data to achieve results comparable to those of the **State-of-the-art (SOTA)** model RoBERTa while using only a quarter of the number of calculations.

To test the validity of pretraining models across languages, Cui et al. presented MacBERT [137] (MLM as a correction) model as a pretrained model. The model improves RoBERTa from several aspects, such as reducing the gap between the pretraining and the fine-tuning phases by masking similar words. It also replaces the NSP pretraining task with the SOP in ALBERT. In addition, BERT has been pretrained on biomedical [48] and clinical [29] data to achieve state-of-the-art performance across biomedical tasks [43].

To balance efficiency and effectiveness, Hofstätter et al. proposed the **transformer-kernel (TK)** model [104, 105], which improves the architecture of the KNRM [18]. The TK model encodes queries and documents independently by using a small number of transformer layers (up to three) before kernel pooling to obtain an optimal ratio between effectiveness and efficiency. Hofstätter et al. also proposed the TK model with local attention [106], which uses local self-attention rather than the self-attention of the encoder of the transformer to perform pooling over windows of document terms via a modified KNRM.

The above-mentioned pretrained language models require different training and fine-tuning phases to adapt to different tasks. By contrast, Raffel et al. introduced a unified architecture, T5 (text-to-text transfer transformer) [25], that converts all NLP tasks into text-to-text tasks. T5 uses more hyperparameters and training data than BERT but allows for the direct use of the same model, objective, training process, and decoding process for every task. However, T5 is a mere combination of BERT and GPT in which the correlation between the encoder and the decoder enables contextualized output embedding as well as a self-attention mechanism that allows each word to attend differently to inputs based on different attention mechanisms.

More recently, transformer decoder-based generative language models, such as GPT-3 [119], ChatGPT [80], GPT-4 [87], **Pathways Language Model (PaLM)** [4, 95], and **Large Language Model Meta AI (LLaMA)** [38, 39], have drawn considerable attention owing to their impressive capability to solve a wide range of tasks without requiring any task-specific fine-tuning. These decoder-based auto-regressive large transformer language models are pretrained on a large amount of self-supervised data, followed by the use of techniques of alignment, like reinforcement learning, based on human feedback to generate humanlike responses. Although these LLMs have demonstrated impressive zero-shot performance, using them in real-world scenarios is challenging due to their consumption of a large amount of computational resources even during inference. Moreover, they can generate responses based only on their pretraining knowledge. This means that they cannot access any information beyond their pretraining timeline. To address these limitations, retrieval-augmented generation has drawn a considerable amount of interest [88]. In this approach, LLMs can retrieve information from external corpora to augment the context of the input. This helps them effectively answer even time-sensitive queries. Specifically, external

knowledge bases store vector embeddings of different documents. Thus, for a given query, the most relevant documents are retrieved based on the similarity between the query embedding and the document embeddings. Due to their effectiveness, embeddings generated by BERT-based models are widely used to retrieve the relevant information via vector similarity.

The relationship between RoBERTa [140], ALBERT [150], ELECTRA [62], MacBERT [137], TK [104, 105], and T5 [25], and the original BERT model [44] suggests that all these models are simply modified versions of the original BERT with adjustable objectives. We called these variants BERT-based models/approaches.

3 BERT FOR AD HOC IR AND CHALLENGES

This section provides a comprehensive survey of state-of-the-art pretrained transformer encoders (i.e., BERT-based approaches) for the application of ad hoc IR and the challenges to them. As BERT is the pioneer in pretrained transformer encoder models, we first briefly present the BERT model introduced by Devlin et al. [44] in Section 3.1. For the purpose of analysis, all BERT-based IR approaches are divided into several categories, and the challenges to each are presented in Sections 3.2, 3.3, and 3.4. Section 3.2 provides a comprehensive overview of BERT-based approaches that handle long documents. Considering the importance of integrating semantic information, Section 3.3 reviews recent BERT-based approaches from different perspectives on contextualization, namely, methods of aggregating long documents, ranking strategies, and weak supervision, in Section 3.3.1, Section 3.3.2, and Section 3.3.3, respectively. We then provide a comparative analysis of these perspectives in Section 3.3.4. Section 3.4 surveys BERT-based approaches that deal with effectiveness and efficiency by focusing on two aspects: reducing the duration of interactions (in Section 3.4.1) and reducing the number of hyperparameters (in Section 3.4.2). In addition, Section 3.4 contains Section 3.4.3, which summarizes BERT-based approaches and considers effective training strategies before a comparative study is provided in Section 3.4.4. We also introduce a category of BERT-based term weight prediction in Section 3.5. Finally, we present a review of approaches to BERT-based query expansion and document expansion in Section 3.6 and Section 3.7, respectively.

3.1 Preliminary Overview of BERT

BERT [44] was introduced by Devlin et al. to develop a robust language model. Before the success of BERT, most of the state-of-the-art approaches (e.g., **Contextualized word Vectors (CoVe)** [19], ELMo [73], and GPT [10]) were unidirectional or sequential and thus failed to leverage contextual relationships across text inputs that led to a poor pretrained model. The bidirectional property of the BERT model allows it to learn the context of unlabeled text from both the left and the right sides such that it yields a good pretrained NLP model. BERT is essentially the encoder part of a transformer with the additional step of masking some of the input tokens and the additional objective of predicting the masked word based only on the context of the tokens surrounding it. The encoder receives each of the masked tokens that are conditioned on the remaining tokens in the sentence, where this renders the decoder redundant.

BERT provides both simple and complex models: BERT_{Base} (layers = 12, hidden dimensions = 768, self-attention heads = 12, parameters = 110M) and BERT_{Large} (layers = 24, hidden dimensions = 1,024, self-attention heads = 16, parameters = 340M). Owing to its more complex architecture and a larger number of hyperparameters, BERT_{Large} outperforms BERT_{Base} under the same training and fine-tuning processes. The input representation of BERT consists of three parts: a WordPiece token embedding [142] (i.e., the original vocabulary consists of individual symbols in the language, and the most frequent combinations of symbols are then iteratively added to the vocabulary), position embedding, and segment embedding (each token of the first sentence is denoted by E_A , and each

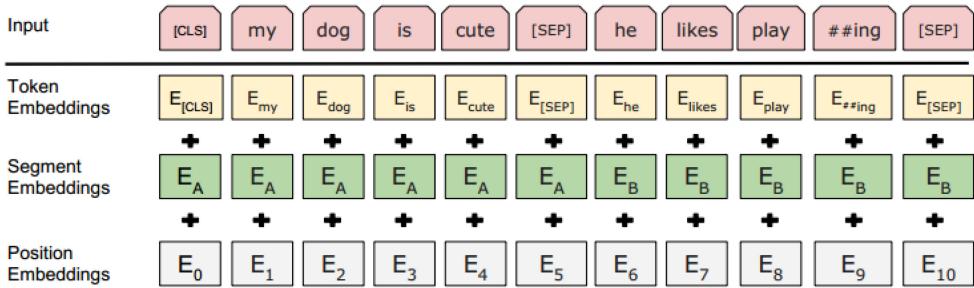


Fig. 2. The input representation of BERT consists of three parts: token embeddings, segment embeddings, and position embeddings. The special token [CLS] is always the first token in every sequence, and the special token [SEP] separates two sentences in it [44].

token of the second sentence is denoted by E_B (see Figure 2 for details). There are two special tokens in BERT: the [CLS] token and the [SEP] token. The [CLS] token is always the first token of the input sequence for classification and the [SEP] token is the separator between sequences. The maximum length of an input containing the two special tokens is limited to 512.

BERT addresses the limitations and disadvantages of ELMo and OpenAI GPT by using the pre-training objectives of *MLM* and *NSP*. The *MLM* randomly masks some words from the input texts during training and then predicts them from contextual information. During pretraining, 15% of WordPiece (i.e., a variant of byte pair encoding [101] methods) [142] tokens were chosen at random for prediction. Each chosen token had a 80% chance of being replaced by the [MASK] token, a 10% chance of being replaced by a random token, and a 10% chance of self-replacement. In addition, BERT uses *NSP* to predict whether a pair of sentences is sequential (i.e., whether the second sentence is the next sentence in the original text).

3.2 Utilizing BERT for Handling Long Documents

As a pioneer pretrained language model, BERT [44] has drawn significant attention from the NLP community. Although BERT [44] is successful on short text-based applications (usually limited to 512 tokens), it fails to process long documents due to the limitation on the maximum length of the input sequence. To resolve this issue, state-of-the-art BERT-based methods rely on two categories of models: the aggregation-guided model and the block selection model.

State-of-the-art aggregation-guided methods [74, 131, 143, 146] focus on dividing a long document into small segments, e.g., sentences and overlapping passages, where the final document score is an aggregation of the top k scores retrieved by BERT and the original document score retrieved by a traditional model. Section 3.3.1 provides a comprehensive review of aggregation-based approaches to BERT. It suggests that such approaches may suffer from a loss of important information that leads to problems related to time, memory, and energy consumption.

Different from aggregation-guided methods, the block selection model (e.g., KeyBLD [74]) relies on selecting key blocks to form a short document that can then be processed by using a state-of-the-art BERT-based model. The idea of selecting key blocks with respect to the entire document by using the BERT model was proposed by Ding et al. in **Cognize Long TeXts (CogLTX)** [71]. Inspired by the cognitive theory proposed by Baddeley [3], CogLTX considers the analogy between BERT and human memory to implement long-term memory. The main step in CogLTX is MemRecall, the process of identifying the relevant text blocks by treating the blocks as episodic memories. MemRecall imitates the working memory in terms of competitive retrieval, rehearsal, and decay

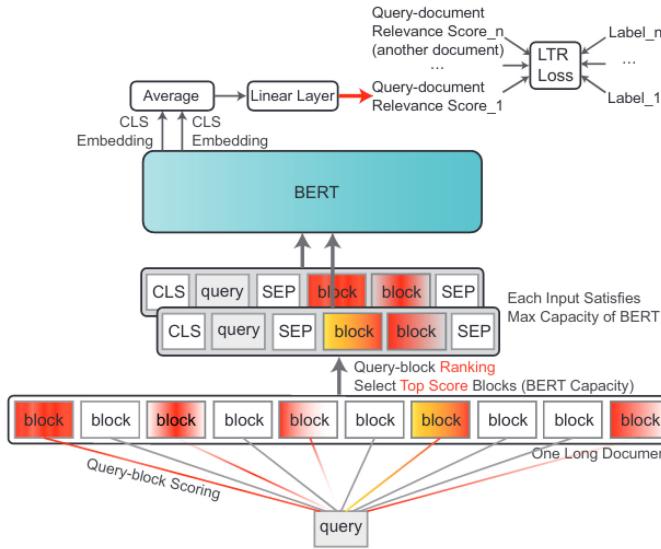


Fig. 3. Overview of the architecture of KeyBLD. [74]. The KeyBLD model comprises four key components: block segmentation, block selection, query-blocks representation, and document ranking. The block segmentation phase divides documents into smaller blocks, while traditional IR models assign relevance scores to each block. The query-blocks representation step concatenates the query with the most relevant blocks, and this is the input to the BERT model for document re-ranking and improved retrieval performance.

to facilitate multi-step reasoning. Another BERT model, called judge, is introduced to CogLTX to score the relevance of the blocks and is trained jointly with the original BERT reasoner. Although the use of CogLTX for IR applications has not been explored, it works as a basic building block for KeyBLD [74], which is used in IR applications. In contrast to CogLTX, KeyBLD [74] first selects the key blocks of a long document by local query-block preranking and then aggregates a few blocks to form a short document that can be processed by BERT for ad hoc IR. Figure 3 illustrates the basic architecture of KeyBLD. The experimental analysis of KeyBLD [74] indicates that it outperforms the relevant state-of-the-art approaches and is faster than them.

3.3 Utilizing BERT for Integrating Semantic Information

The bidirectional property of BERT-based models enables the learning of the context (i.e., semantic information) of unlabeled texts from both the left and the right sides and lays the foundation for obtaining contextualized embeddings. We first briefly discuss the use of BERT-based models for aggregating long documents in Section 3.3.1 and then present ranking strategies in Section 3.3.2. We then cover techniques of weak supervision that leverage BERT-like models in Section 3.3.3. Finally, we provide an extended comparison and analyses based on experiments in Section 3.3.4.

3.3.1 Aggregation Methods for Long Documents. A number of state-of-the-art approaches [74, 131, 143, 146] can be considered to be aggregation techniques that combine shorter segments of texts to handle long documents. Aggregation-guided BERT models for IR can be divided into three groups, sentence-level score aggregation, passage-level score aggregation, and passage-level representation aggregation, as shown in Table 1. This section briefly discusses aggregation-guided BERT-based approaches as a subclass of BERT-based models for integrating semantic information, while the previous section (Section 3.2) considered aggregation-guided methods and block selection models for handling long documents.

Table 1. Aggregation Methods for Handling Long Documents

Type	Model	Aggregation Methods
Sentence-level score aggregation	[131, 143]	Top-3
Passage-level score aggregation	[146]	FirstP, MaxP, SumP
	[135]	AvgP
	[107]	TopKP
Passage-level representation aggregation	[111]	AvgP
	[23]	PARADE _{Avg} , PARADE _{Sum} , PARADE _{Max} , PARADE _{Attn} , PARADE _{CNN} , PARADE _{Transformer}

- Sentence-level score aggregation: The state-of-the-art [131, 143] approaches of this sort divide the document into a set of sentences, where the final document score is a linear aggregation of the top k scores retrieved by BERT and the original document score retrieved by the traditional model. Yang et al. [131] used the sentence-level model to apply BERT to document retrieval and divided the document into a set of sentences by combining the BERT score with the BM25 (Anserini [93]) score to obtain the final score of document retrieval. Specifically, the final document score was a linear aggregation of the top k scores retrieved by BERT and the original document score retrieved by the traditional model. Yilmaz et al. [131] proposed a system called *Birch* [143], which is similar to the proposal in Reference [131], which is shown in Figure 4. Both these studies [131, 143] reported experiments on two datasets: the TREC 2011-2014 Microblog and Robust04. The lengths of the documents in the TREC Microblog dataset fitted the BERT model well. Every document in the Robust04 dataset was split into a set of sentences. The results on the TREC Microblog dataset showed that BERT outperformed previously developed neural ranking models and demonstrated its suitability for passage ranking. The experiments on the Robust04 dataset involved a linear aggregation of the scores of the top three sentences and the original document score as the final document score. The results of Birch show that fine-tuning across multiple datasets can help obtain a better fit with the target dataset. For example, Birch (MS → MB) achieved the best results and outperformed Birch fine-tuned on **Microsoft Machine Reading COnprehension (MS MARCO)**, while Birch (MS → MB) was first fine-tuned on MS MARCO (MS) and then on the TREC Microblog (MB) dataset. However, the sentence-level score aggregation-based BERT suffers from the loss of important information as well as relevant judgment.

- Passage-level score aggregation: The state-of-the-art approaches [107, 135, 146] of this kind rely on segmenting long documents into small overlapping passages. These passages are formed by considering the passage length and the stride length to prevent passages of a long document from becoming independent of one another. Hence, document-level retrieval can be realized by aggregating passage scores [107, 135, 146]. Dai and Callan proposed different aggregation scores that split a long document into overlapping passages of length 150, i.e., a 150-word sliding window with a stride of 75 words [146]. Any of three approaches can be used to obtain the final document-level retrieval score: (1) BERT-FirstP, by selecting the score of the first passage; (2) BERT-MaxP, by selecting the maximum passage score; and (3) BERT-SumP, by taking the sum of all passage scores. Zhang et al. proposed AvgP [135], which takes the average score of all passages as the final document retrieval score. MaxP and TopKP must obtain all passage relevance scores via an expensive pretrained model, e.g., BERT, and select the maximum relevant passage or the top k passages. To simplify the selection process, Hofstätter et al. selected the top k passages through a lightweight and fast model [107] called the efficient student model and estimated the selected passages by using an expensive model. Despite its success, this strategy may still cause problems related to time, memory, and energy consumption. Furthermore, considering passages irrelevant

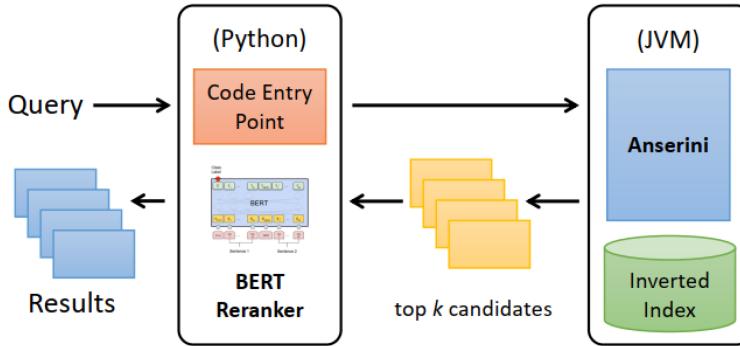


Fig. 4. The architecture of Birch showcases seamless integration between Python and the Java Virtual Machine (JVM), thus enabling the efficient utilization of neural networks like BERT. The main code-entry point implemented in Python leverages the capabilities of the JVM to facilitate document retrieval through the Lucene search library. The top k candidates retrieved from JVM's Anserini are then input to BERT for a secondary ranking process [143].

to the given query may introduce noise to the final representation and limit the identification of long-distance dependencies between the relevant tokens.

- Passage-level representation aggregation: State-of-the-art approaches [23, 111] that fall into this category rely on aggregating the passage representations rather than the passage scores. Li et al. proposed **passage representation aggregation for document reranking (PARADE)** [23]. It uses six approaches to aggregate passage-level signals for a long document: (1) PARADE_{Avg}, which adopts average pooling for the passage representations of a long document to obtain the final document representation, where this is similar to **contextualized embeddings for document ranking (CEDR)** [111]; (2) PARADE_{Sum}, which performs additive pooling across passage representations; (3) PARADE_{Max}, which uses element-wise max pooling for all passage representations to obtain the representation of the final document; (4) PARADE_{Attn}, which adopts a feedforward network to generate an attention weight for each passage; (5) PARADE_{CNN}, which generates pairs of passage representations using several CNNs until only one remains; and (6) PARADE_{Transformer}, which aggregates passage representations through two randomly initialized transformer encoders. To obtain context-aware passage-level signals, Wu et al. introduced the passage-level cumulative gain model to rank documents [153]. In their model, BERT was first applied to obtain passage-level [CLS] representations, following which they were aggregated via LSTM [5].

3.3.2 Ranking Strategies via BERT. To investigate how pretrained transformer encoder-based contextualized language models can be utilized for ad hoc document ranking, researchers have attempted to retrieve documents via various ranking strategies. As shown in References [98, 131, 143], BERT-like models have a better capability of representation than state-of-the-art non-BERT-based IR approaches [17, 18, 46, 89, 116, 125] for ad hoc document ranking. MacAvaney et al. introduced CEDR [111], which combines BERT embeddings with neural ranking models to enhance term representations and improve the performance of neural ranking models. They combined the contextualized word embeddings of BERT and ELMo with certain neural ranking models, including the position-aware deep model for relevance matching in IR: **Position-Aware Convolutional Recurrent Relevance Matching (PACRR)** [63], KNRM [18], and DRMM [46]. The BERT model can simultaneously encode multiple text segments by utilizing the two meta-tokens ([SEP] and [CLS]) and text segment embeddings (e.g., Segments A and B). [SEP] acts as a separator of each

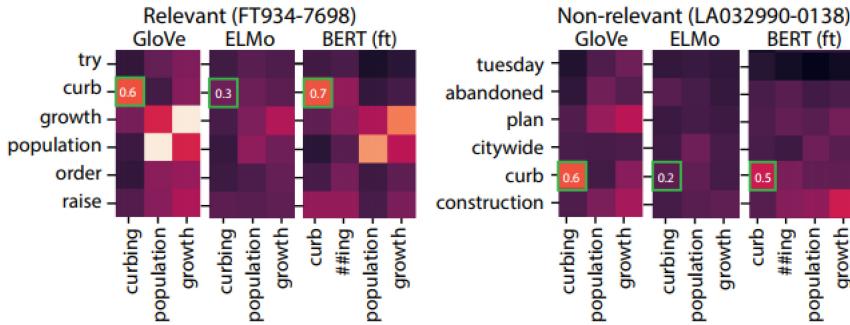


Fig. 5. Comparison of GloVe word embeddings, ELMo representations (layer 2), and fine-tuned BERT representations (layer 5) for relevant and irrelevant documents in relation to a given query [111]. Lighter colors represent higher similarity scores.

segment, while [CLS] is used to make contextual judgments about the text pairs. More precisely, [CLS] is used during training to predict whether two segments are sequential, e.g., whether Segment A immediately precedes Segment B in the original text. Thus, incorporating the [CLS] token into the neural network allows the neural rankers to benefit from deep semantic information from BERT in addition to individual contextualized token matches. Incorporating the [CLS] token into the ranking models is a two-step process. First, the model produces relevance scores for each query term based on similarity matrices. Then, for models using a dense combination, CEDR concatenates the [CLS] vector with the signals of the model. For models that sum query term scores, such as the DRMM [46], CEDR shows that using the [CLS] vector in the dense calculation is effective. To better understand the contributions of contextualized word embeddings to the model, the experimental evaluation of CEDR involves visual comparison of GloVe word embeddings, ELMo representations (layer 2), and fine-tuned BERT representations (layer 5) as shown in Figure 5. For the query “curbing population growth” in this figure, the term “curb” that appears in both relevant and non-relevant documents has two distinct senses. However, GloVe predicts the same similarity score for both candidates. By contrast, both ELMo and BERT assign a higher similarity score to the appropriate sense of the term in response to the query, demonstrating the capabilities of semantic understanding of contextualized models. Their performance on the Robust04 and Web-Track 2012-14 datasets demonstrated that the neural ranking models that used fine-tuned BERT representations (CEDR-PACRR, CEDR-KNRM, and CEDR-DRMM) outperformed the others. This verified the influence of contextualized embeddings on the performance of neural ranking models.

Qiao et al. [88] systematically analyzed the [CLS] embedding in different layers of BERT. In their work, three interaction-focused models based on the BERT_{Large} model were introduced: BERT_{Last-Int}, BERT_{Mult-Int}, and BERT_{Term-Trans}. BERT_{Last-Int} extracts features from the last layer of the [CLS] embedding and then combines them with a weight parameter. Inspired by BERT_{Last-Int}, BERT_{Mult-Int} extracts features from the [CLS] embeddings of all layers. Each layer’s [CLS] embedding is first combined with a weight parameter, and the features of all layers are then integrated. BERT_{Term-Trans} produces a transition matrix for the query and the document and then integrates the transition matrices from all layers via mean pooling. Qiao et al. [88] conducted experiments on two datasets: MS MARCO and ClueWeb09-B. The results showed that the three interaction-based BERT models delivered varying results on different datasets. BERT_{Last-Int} delivered better performance than the other two models on the MS MARCO dataset, and BERT_{Mult-Int} delivered the worst performance in terms of MRR@10. However, BERT_{Mult-Int} outperformed the other two methods

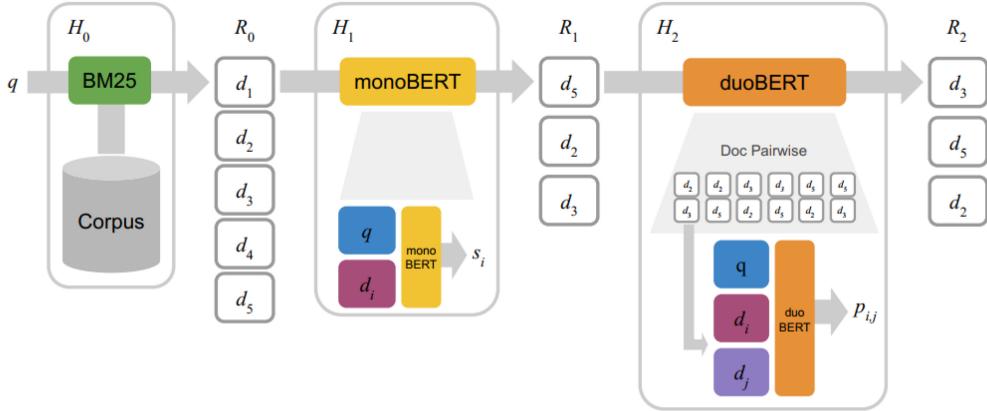


Fig. 6. Architecture of the multi-stage ranking model proposed by Nogueira et al. [99]. The process consists of three stages, denoted by H_0 , H_1 , and H_2 . The first stage, H_0 , produces an initial ranking list retrieved by a simple traditional model for the entire corpus. The second stage, H_1 , monoBERT, serves as a reranker to obtain a more precise ranking. In the third stage, duoBERT adopts pairwise learning to further enhance the quality of the ranking list.

on CluWeb09-B in terms of **Normalized Discounted Cumulative Gain values at a cutoff of 20 (NDCG@20)** and **Expected Reciprocal Rank at position 20 (ERR@20)**.

All the important models discussed above examine correlations between documents and a query to obtain a final, sorted list related to the query. This raises the following question: If all elements of a previously sorted list are fed in pairs into BERT for comparison and a second sorted list is thus obtained, then is the latter list consistent with the former list? Nogueira et al. provided an answer in Reference [99]. They used three stages, denoted by H_0 , H_1 , and H_2 , in an iterative process (i.e., the output of each stage served as the input to the next stage). The first stage, H_0 , produced an initial ranking retrieved by a simple traditional model for the entire corpus that was denoted by R_0 (i.e., the top k_0 candidate documents). In the second stage, H_1 , monoBERT [99], served as a reranker to obtain a more precise ranking R_1 from R_0 . Finally, the top k_1 candidate documents of R_1 were fed into duoBERT [99] in the third stage H_2 . Furthermore, duoBERT used a triple (q, d_i, d_j) as input to produce the probability $p_{i,j}$ used to assess the degree to which candidate d_i was more relevant than d_j , where $d_i, d_j \in R_1$. The architecture of this system is illustrated in Figure 6. The duoBERT model can be characterized as a pairwise approach and monoBERT as a pointwise approach in this framework. The authors conducted experiments on the MS MARCO and TREC CAR datasets and represented the results in terms of the **mean reciprocal rank (MRR)** and the **mean average precision (MAP)** during different stages of retrieval. The experimental findings indicated that a multi-stage architecture offers more advantages in generating an accurate ranking list than a single-stage architecture. The primary limitation of the single-stage architecture lies in its inability to learn the relationships between documents. By contrast, the multi-stage architecture addresses this challenge by framing the problem of document ranking as a comparison of different documents. This approach is crucial for achieving a high-quality ranking list. While *monoBERT* and *duoBERT* apply different ranking strategies in different stages, *duoBERT* can more precisely find the most relevant document in a pair but incurs a higher computational cost.

3.3.3 Weak Supervision via BERT. While supervised learning technology has achieved great success, it has come at the cost of expensive data annotation. Like many deep learning architectures, the data-hungry nature of BERT poses a significant challenge to accurate retrieval. Dai and

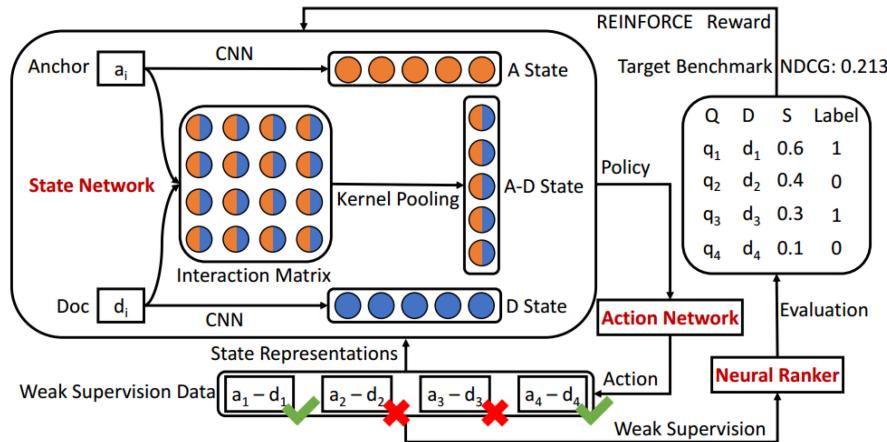


Fig. 7. The architecture of ReInfoSelect [65] consists of a state network, an action network, and a reward. The state network represents the anchor-document (a-d) pair while the action network determines whether to retain the pair. The training reward is obtained from the ranker that has undergone training.

Callan [146] found that when BERT was fine-tuned on user clicks from Bing, its accuracy on TREC Web Tracks improved by 16% in comparison with the scenario when only masked LM-based pre-training was used. Weakly supervised pretrained models are trained on extensive amounts of data with limited or incomplete labeling information, leading them to acquire more general features and semantic representations. Consequently, these models exhibit impressive performance on new domains or tasks. On the contrary, the efficacy of a supervised model on a specific task may be constrained by the domain of its training data.

Although a number of weak supervision-based ad hoc approaches to IR are available [24, 40, 41, 70, 82, 83, 102], few studies [65, 92] have considered a BERT architecture that relies on weak supervision. Zhang et al. proposed a reinforcement learning technique, *ReInfoSelect* [65], that can guide ranking models to obtain more useful information. It consists of three components, a state network, an action network, and a training reward, as illustrated in Figure 7. The state network contains an **anchor-document (a-d)** pair that is fed into the action network to determine whether it should be used. ReInfoSelect uses the *policygradient* approach to connect the state network to the action network to select useful a-d pairs for weak supervision. The remaining a-d pairs are fed into BERT to obtain a reward and then are returned to the state and action networks. Their normalized discounted cumulative gain values at a cutoff of 20 (as in NDCG@20) and ERR@20 on the ClueWeb09-B, Robust04, and ClueWeb12-B13 datasets, obtained by using both Conv-KNRM and BERT, were then compared. The results indicated good performance. Xu et al. [92] proposed weak supervision for passage ranking via BERT. This model consists of three components: labeling functions, label aggregation, and supervised training. The labeling functions consist of four scoring functions: (1) BM25 score, (2) **Term Frequency-Inverse Document Frequency (TF-IDF)**, (3) cosine similarity of the representation of a universal embedding [26], and (4) cosine similarity of the last hidden layer. The most significant difference between the weakly supervised methods was in the manner in which the labels were obtained. To simplify the process of interaction, Hofstätter et al. used the top k passages via a lightweight and fast selection model, the **Efficient Student Model (ESM)** [108]. They then used a computationally expensive model to estimate the relevance scores of the selected passages, rather than making relevance estimations for all passages of a document, to avoid wasting computational resources.

Table 2. Comparison of Empirical Results of Different Methods in Terms of MRR@10, MAP, and NDCG@20 on the MS MARCO, TREC CAR, and Robust04 Datasets

Models		MS MARCO dataset		TREC CAR dataset	Robust04 dataset
Strategy	Approach	MRR@10 (Dev)	MRR@10 (Eval)	MAP (Test)	NDCG@20
Traditional IR model	BM25 (Microsoft Baseline) BM25 (Anserini)	0.167 0.187	0.165 0.190	0.130 0.153	0.414 —
The different architectures of BERT	<i>BERT_{Base}</i> [98] <i>BERT_{Large}</i>	0.347 0.365	— 0.358	0.310 0.335	— —
Models based on BERT_{Base}					
Passage-level representation aggregation via BERT	PARADE-Avg [23]	—	—	—	0.512
	PARADE-Sum	—	—	—	0.539
	PARADE-Max	—	—	—	0.544
	PARADE-Attn	—	—	—	0.527
	PARADE-CNN	—	—	—	0.563
	PARADE-Transformer	—	—	—	0.566
Combining BERT with neural ranking models	CEDR-PACRR via BERT [111]	—	—	—	0.515
	CEDR-KNRM via BERT	—	—	—	0.538
	CEDR-DRMM via BERT	—	—	—	0.526
Weak supervision via BERT	ReInfoSelect (Conv-KNRM) [65]	—	—	—	0.442
	ReInfoSelect (BERT)	—	—	—	0.450
Models based on BERT_{Large}					
Multi-stage ranking via BERT	(Anserini BM25) monoBERT [99]	0.372	0.365	0.348	—
	monoBERT + <i>duoBERT_{MAX}</i>	0.326	—	0.326	—
	monoBERT + <i>duoBERT_{MIN}</i>	0.379	—	—	—
	monoBERT + <i>duoBERT_{SUM}</i>	0.382	0.370	0.369	—
	monoBERT + <i>duoBERT_{BINARY}</i>	0.383	—	0.369	—
	monoBERT + <i>duoBERT_{SUM}</i> + TCP (target corpus pretraining)	0.390	0.379	—	—
Constructing various contextualized embeddings via BERT	BERT (Last-Int) [88]	0.337	0.359	—	—
	BERT (Mult-Int)	0.306	0.329	—	—
	BERT (Term-Trans)	0.331	0.356	—	—

3.3.4 Extended Analysis of Comparisons and Experiments. In light of differences in the datasets and the evaluation indicators used in different studies, their experimental comparison is provided in Table 2. We considered representative models and compared their values in terms of different evaluation metrics on the MS MARCO, TREC CAR, and Robust04 datasets. All the considered approaches [65, 88, 98, 99, 111] used consistent experimental settings that enabled us to draw a fair comparison. For example, for the MS MARCO dataset, all models in Table 2 utilized the official BERT models that had been pretrained on the full Wikipedia website, while for the TREC CAR dataset, the BERT reranker was pretrained on half of the Wikipedia website to avoid the leakage of test data into the training data.

In Table 2, both BERT_{Base} and BERT_{Large} serve as the reranker to rerank candidate documents retrieved by the traditional model, e.g., BM25. The above comparison yields three main conclusions:

- The MS MARCO dataset has an average of one relevant passage per query. To estimate the performance of the models on MS MARCO, the researchers used the mean reciprocal rank of the first correct answer. Because Anserini [93] aims to connect academic research on IR with the development of search applications, the initial candidate list produced by Anserini’s implementation of the BM25 algorithm exhibited a notable improvement of 12% over the BM25 implementation by Microsoft on the MS MARCO dataset. This distinction had a noticeable impact on the final retrieval performance, specifically when comparing monoBERT with the BERT models introduced by Nogueira et al. [98]. In this context, monoBERT utilizes BERT as the binary relevance classifier. This has also been illustrated by Lin et al. [50]. Moreover, monoBERT + duoBERT_{BINARY} and monoBERT + duoBERT_{SUM} outperformed monoBERT + duoBERT_{MIN} on the development

set. By using the *TCP*, monoBERT + duoBERT_{SUM} achieved the best performance. monoBERT + duoBERT_{MAX} delivered the worst performance on the development set, 6% worse than that of BERT_{Base}. In addition, BERT_{Base} outperformed BERT_{Last-Int}, BERT_{Multi-Int}, and BERT_{Term-Trans}. Furthermore, Nogueira et al. applied the [CLS] embedding of the last layer, which is similar to that of BERT_{Last-Int}. However, the variable gap in performance among BERT_{Base}, BERT_{Last-Int}, BERT_{Multi-Int}, and BERT_{Term-Trans} might have stemmed from the variable batch size. In general, it seems that applying BERT to obtain the correlation between candidate documents yields a significant performance improvement, which in turn yields a novel area for sorting tasks in the field of IR.

- Similarly, monoBERT + duoBERT_{BINARY} and monoBERT + duoBERT_{SUM} delivered the best performance on the TREC CAR dataset, while monoBERT + duoBERT_{MAX} performed the worst, about 3% lower than BERT_{Large}.

- The original DRMM was used to compute the cosine distance between each word embedding of the query and all word embeddings of documents in the Robust04 dataset, and then they were mapped them to a histogram. It can thus be considered a correlation model that omits location information. However, PACRR and KNRM can extract local contextualized information through their convolution kernels. Using the contextualized word embeddings of BERT, CEDR-KNRM achieved the best performance in terms of NDCG@20 on Robust04, and CEDR-PACRR performed slightly worse than CEDR-DRMM. The performance of PARADE-CNN and PARADE-Transformer was superior to that of the variants of CEDR, demonstrating that hierarchical aggregations can better preserve information that is beneficial to document-ranking. Unsurprisingly, ReInfoSelect (Conv-KNRM) and ReInfoSelect (BERT) performed slightly better than BM25.

3.4 BERT for Balancing Effectiveness and Efficiency

BERT models are more effective than previously proposed neural network ranking models in terms of document retrieval. However, BERT_{Base} has 110 million parameters and BERT_{Large} has 340 million parameters. Therefore, if the query and the text are fed simultaneously into BERT, then its computational complexity is high, leading to a long computation time. To eliminate this deficiency, researchers have focused on two aspects: (1) reducing the interaction time between the query and the document, i.e., through dual-encoder-based models, and (2) reducing the number of hyperparameters of BERT, i.e., through distillation models. In light of these aspects, we discuss the state-of-the-art dual-encoder-based models, distillation models, and effective training strategies in Sections 3.4.1, 3.4.2, and 3.4.3, respectively. Finally, we provide an extended analysis of the relevant comparisons and experiments in Section 3.4.4.

3.4.1 Dual-encoder-based Models. Researchers have recently attempted to retrieve documents in a low-dimensional embedding space [58, 69] through a dense retrieval model, the architecture of which is based on the representation-focused model. It maps queries and documents into low-dimensional embeddings via dual-encoders (i.e., the query encoder and the document encoder).

To achieve a better tradeoff between effectiveness and efficiency, researchers have attempted to encode independent queries and documents to reduce the computational complexity of the stage involving their interaction. Qiao et al. proposed BERT(Rep) [88], which independently encodes the query and the document via BERT and uses the [CLS] embeddings of the last layer as the representations of the query and the document. Then, relevance scores are calculated based on their cosine similarities. They used the MS MARCO and CluWeb09-B datasets for their experiments. ERR@10 was used for evaluation on the MS MARCO dataset while NDCG@20 and ERR@20 were used for evaluation on CluWeb09-B. BERT (Rep) was trained on “Train Triples” provided by MS MARCO for its dataset.

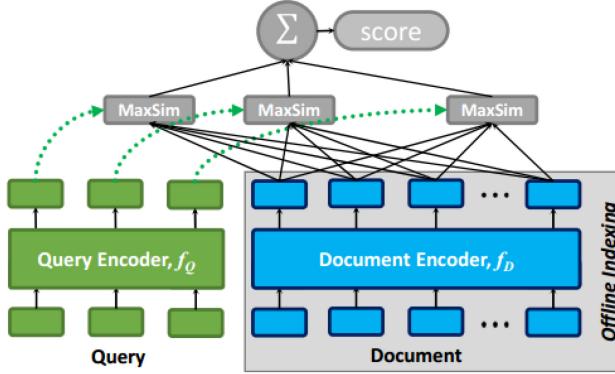


Fig. 8. ColBERT [86] independently encodes queries and documents via BERT and can reduce the computational complexity of the model through late interaction.

BERT-based dense models of retrieval focus on dense passage retrieval [58, 85, 86, 115, 141]. Khattab and Zaharia proposed a post-interaction architecture called ColBERT [86] that independently encodes queries and documents via BERT to reduce the computational complexity of the model ($n^2 \rightarrow n$) and indexes the documents offline to reduce its time cost. ColBERT contains three components, a query encoder (f_Q), a document encoder (f_D), and an interaction machine, all of which are shown in Figure 8. ColBERT provides a novel approach to identifying the interactions between a query and a document. The query and the document are first encoded separately via BERT to store the word embeddings, and the similarity between each query embedding and all document embeddings is then calculated via an interaction machine to obtain the maximum score. Finally, all the maximum scores of the query–document token pairs are integrated to determine the relevance between the query and the document. The authors conducted experiments on the MS MARCO and the TREC CAR datasets and provided a comparison with BERT_{Base} on the reranking and the end-to-end tasks in terms of MRR@10. BERT_{Base} was assessed in terms of MAP and MRR@10 values on the TREC CAR dataset. Based on the experimental results, ColBERT achieves better performance than BERT (Rep).

ColBERT [103] was further improved by Lin et al. [50] by using knowledge distillation. Reimers and Gurevych introduced Sentence-BERT [85] for passage ranking to obtain sentence embeddings via the Siamese network. Similarly to CEDR [111] and PARADE_{Avg}, RepBERT [58] executes average pooling for all tokens in a passage to produce its final representation. DPR [123] trains the query encoder and the document encoder on natural questions, Triviaqa, web questions, CuratedTrec, and Squad V1.1.

3.4.2 Distillation Models. To address the limitations related to memory and the long time needed by state-of-the-art BERT models to make inferences, researchers have developed distillation models of BERT. Jiao et al. proposed TinyBERT [132] that uses transformer distillation, general distillation, and task-specific distillation. Transformer distillation relies on learning attention weights to enable the teacher (BERT) to transfer its linguistic knowledge to the student (TinyBERT), while the other two steps of distillation subsequently utilize the learned knowledge to perform task-specific distillation on augmented datasets. In contrast to TinyBERT, which uses distillation for building task-specific models, Sanh et al. proposed DistilBERT [124], which uses knowledge distillation in pretraining to develop a lightweight, general-purpose, and pretrained model. By utilizing advanced optimization techniques and reducing the number of layers, DistilBERT [124] makes it possible to attain similar performance to that of larger models on

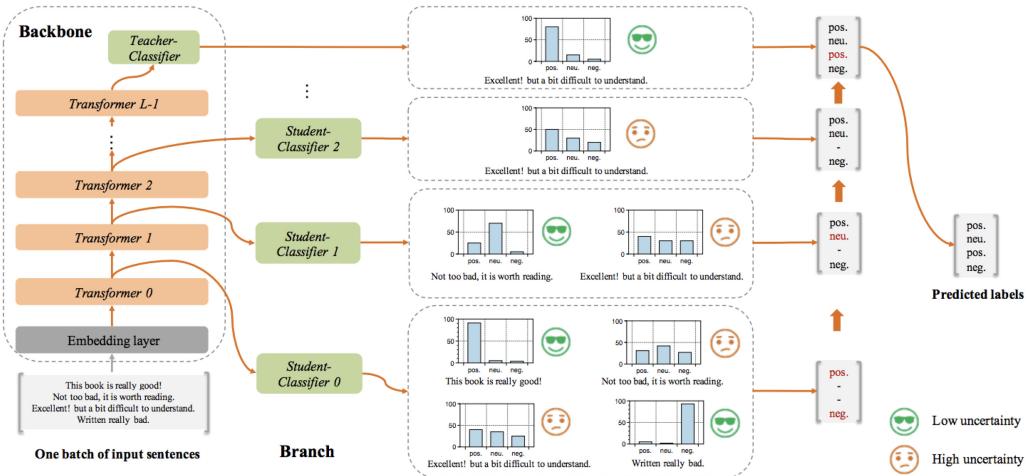


Fig. 9. The inference procedure of FastBERT [127]. Given a batch of input sentences, the embedding layer translates it into an embedding sequence. Transformer0 and Student-classifier0 infer its labels as probability distributions while calculating uncertainty.

many downstream tasks by using much smaller language models pretrained by using knowledge distillation. Unlike in References [124, 132], FastBERT [127] uses a unique self-distillation mechanism during training and an adaptive mechanism for inferences to improve its efficiency without compromising its performance. Furthermore, FastBERT incorporates a dynamic layer adjustment mechanism to reduce computational complexity. This means that the model can adaptively modify the number of execution layers for each sample. For simpler samples, FastBERT utilizes fewer layers for prediction, while employing deeper layers for more intricate samples. Moreover, FastBERT incorporates a distinctive self-distillation processing mechanism that enables faster and more accurate results while minimizing changes to the model structure. The inferential process of FastBERT is illustrated in Figure 9. In contrast to the above distillation-based variants of BERT, Lan et al. proposed the ALBERT [150] model, which has a significantly smaller size and improves the speed of inferences. Note that ALBERT mainly changes the model architecture through factorized embedding parameterization, cross-layer parameter sharing, and inter-sentence loss of coherence.

To reduce the temporal complexity of inference in the two-input case from quadratic to linear, TwinBERT [128] uses twin-structured BERT-like encoders to encode the query and the document, respectively, and a crossing layer to combine the two embeddings and produce a similarity score. In contrast to BERT, the document representations in it are precomputed and stored via *offline caching*. Due to the success of the knowledge distillation technique, the training of TwinBERT is based on the teacher–student framework.

3.4.3 Effective Training Strategies. Researchers have explored different training strategies for pretrained transformer encoder-based contextualized language models. For instance, **Sentence-BERT (SBERT)** [85] uses Siamese and triple networks to generate contextual sentence embeddings. It uses cosine similarity or the Manhattan/Euclidean distance in the interaction stage to find semantically similar sentences. SBERT reduces the computational complexity of the system from n^2 to n while ensuring accuracy. **Approximate nearest-neighbor negative contrastive learning (ANCE)** [69] constructs negative samples from the corpus via the approximate nearest neighbor algorithm, where the total number of negative samples is dynamically changed based

Table 3. Comparison of Empirical Results in Terms of MRR@10 on MS MARCO

Models		MS MARCO Passage Retrieval Dataset	
Strategy	Approach	MRR@10 (Dev)	MRR@10 (Eval)
Traditional IR model	BM25 (Microsoft Baseline)	0.167	0.165
	BM25 (Anserini)	0.187	0.190
Dual-encoder-based models	RepBERT [58](over BERT _{Base})	0.304	0.294
	ColBERT (over BERT _{Base}) [86]	0.349	0.349
	DPR (BM25 + Rand Neg)(over BERT _{Base}) [123]	0.311	—
Effective training strategies	ANCE (FirstP)[69](over RoBERTa Base)	0.330	—
	TSA (Pairwise + In-Batch)[107](over BERT _{Base})	0.338	—
	TSA-Balanced (Pairwise + In-Batch)	0.340	—
Enhancing retrieval process with term weights	DeepCT [147]	0.243	—
	EPIC + BM25 [110]	0.273	—
	SparTerm [136]	0.275	—
	SPLADE [120]	0.322	—
	DeepImpact [9]	0.326	—
	DeepImpact + ColBERT [9]	0.362	—

on the training samples used. It solves the issue of a difference in distributions between the training and the test datasets. The experimental results on MS MARCO and TREC DL showed that ANCE is superior to models that randomly sample negative values. The **Topic-aware Sampling (TSA)** [107] approach performs a clustering operation by using k -means for queries. Its training can be implemented according to the sampled topics. As a further improvement, Hofstätter et al. used TSA-Balanced to sample the negative values in a batch via the maximum margin. The experimental results on MS MARCO and TREC DL showed that TSA-Balanced is superior to TSA and random sampling. Wang et al. [133] followed the same pipeline that provides representative feedback embeddings via k -means clustering.

3.4.4 Extended Analysis for Comparison and Experiments. Research on representation-focused BERT models has mainly used the MS MARCO passage dataset. Due to its specificity (i.e., an average of one relevant passage per query), researchers use the MRR to obtain the reciprocal rank of the first correct answer. We conducted an experimental comparison of representative models in terms of the MRR10 on the MS MARCO passage dataset (see Table 3).

Table 3 illustrates the superior performance of pretrained BERT models compared with sparse retrieval models, such as BM25. Among the representation-focused BERT models in dual-encoder-based architectures, ColBERT delivers the best performance on the MS MARCO dataset. It leverages token-level contextualized information to match queries and documents, thus enabling more precise targeting of the document content. On the contrary, RepBERT performs poorly on the MS MARCO dataset, as it generates output-contextualized text embeddings through average pooling, which is not advantageous for capturing important representational information. Effective training strategies, specifically TSA-Balanced, outperform other models on the MS MARCO dataset, surpassing both TSA and ANCE. This finding highlights the significant role of negative sample selection in enhancing the capability of the model to discriminate between positive and negative samples. In conclusion, careful consideration of negative sample selection can significantly improve the training process and enhance the model's ability to distinguish between positive and negative instances.

3.5 BERT-based Predictions of Term Weights

IR models consider three major factors in general: the *document length*, *term frequency*, and *term specificity* [16, 54, 60]. However, the importance of a term in a document also depends on the semantic information relating the query to the document [2, 79].

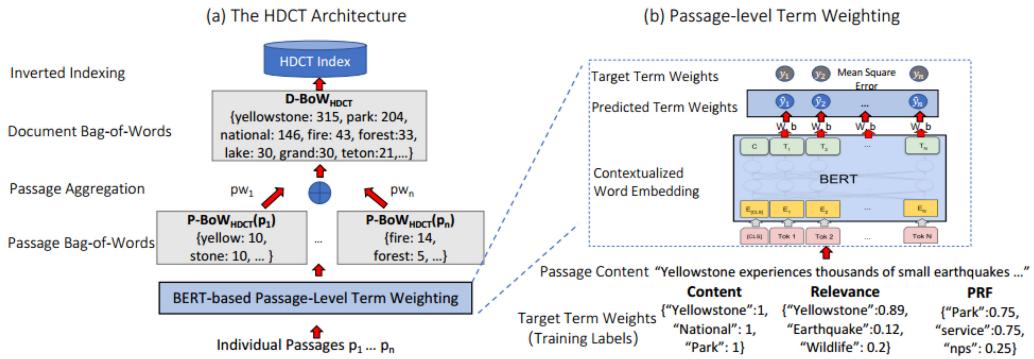


Fig. 10. Architecture of HDCT [148]. The purpose of passage-level term weighting is to obtain the contextualized term weights for each passage. Moreover, HDCT integrates the contextualized term weights for all passages in the document for more accurate document ranking.

To improve the efficiency of the retrieval stage, Dai and Callan proposed context-aware **hierarchical document term weighting (HDCT)** [148] that provides a term-weighting framework for the retrieval stage in document ranking tasks, as illustrated in Figure 10. Given a document d , HDCT first divides it into a set of passages. It produces contextualized word embeddings via BERT for tokens of every passage and then uses a linear layer to obtain a real number as weight. To ensure that integer values are obtained, HDCT uses two-digit precision to scale the predicted weight into an integer, such as the term frequency. The results of experiments showed that HDCT delivers significantly better performance than term frequency.

Dai and Callan also proposed the DeepCT [147] framework that is similar to HDCT. Both use BERT to generate the contextual embedding for each token in the passage and obtain a term frequency-like integer instead of a term frequency value during the retrieval stage. The difference between the frameworks is that the former was designed to rank passages while the latter to rank documents. In contrast to these approaches, MacAvaney et al. proposed **Expansion via Prediction of Importance with Contextualization (EPIC)** [110], which models the importance of query/document terms and executes document expansion. The reranking of passages by EPIC based on lexicon-grounded representations, generated from contextualized language models, helps improve its performance such that it outperforms DeepCT [147], HDCT [148], and DocT5Query [97] (as shown in Table 3). To address limitations in the training of the model as a per-token regression task in DeepCT and HDCT, Mallia et al. introduced DeepImpact [9] that can obtain independent term-based weights regardless of the co-occurrence of terms in the document. The major difference among the above methods is that DeepImpact leverages DocT5Query [97] to enrich the original text via expansion terms.

Most of the above-mentioned studies sought to improve the capacity for dense representations of low-dimension queries (documents), whereas term-based sparse representations play an important role in industrial applications. To improve the sparse representations of Bag of Words models (e.g., BM25), Xiong et al. introduced a framework called **Term-based Sparse representations (SparTerm)** [136] that consists of two components: an importance predictor and a gating controller. The importance predictor can obtain a 305,22D (i.e., the size of BERT's vocab.txt file) passage-wise distribution of importance for each passage by using a pretrained language model, and the gating controller provides expansion-enhanced vectors through binary word vectors of the passage. In a further refinement, Formal et al. proposed **SParse Lexical AnD Expansion model (SPLADE)** [120] on the basis of SparTerm. It adds a logarithmic function to the importance predictor to

prevent the dominance of certain terms and smoothen the distribution of importance throughout the vocabulary. The major difference between SparTerm and SPLADE is in the loss function.

While these models require fine-tuning on task-specific datasets to achieve optimum performance, recently proposed LLMs like GPT-3 have demonstrated the capability of in-context learning via self-supervised pretraining on large amounts of data. Note that in-context learning works based on learning through analogies drawn from the given few-shot examples, thus making it possible for LLMs to solve tasks without requiring any task-specific fine-tuning. However, training only by using self-supervised objectives may still lead to poor performance on many tasks. To address this issue, the use of proximal policy optimization with human feedback in a **reinforcement learning (RL)** framework has been recently proposed. ChatGPT is the latest addition in this series that additionally uses dialog-based instructional data in the supervised and RL-based meta-training stages. ChatGPT has shown the ability to perform numerous tasks and can outperform GPT-3. However, strong zero-shot performance across a wide range of tasks still does not help it achieve optimum performance on IR tasks.

The comparative experimental analysis reported in Table 3 suggests that SparTerm and EPIC (based on BM25) deliver the worst performance and are inferior to SPLADE, which illustrates the importance of utilizing sparse representations. DeepImpact followed by a ColBERT re-ranker is more effective, where this underlines the importance of using independent term-based weighting.

3.6 BERT-based Query Expansion

Document ranking in IR is primarily dependent on the appropriate understanding of the query terms to retrieve the relevant documents. Given the ability of BERT to capture contextual information based on the input text, it is expected that longer queries contain more information than shorter queries. Thus, the same collection of documents may result in a higher accuracy of document retrieval for longer queries. This situation has led to the emergence of the concept of query expansion [57, 155].

To expand queries for reranking by BERT, Padaki et al. [96] proposed three methods: *Expansion with structure*, *Expansion with concepts*, and expansion. The details of these approaches are as follows:

- (1) Expansion with structure produces a coherent sentence by adding structure words to it. The relevant study focused on two main approaches when adding a structure: *GeneratedStructure* and *TemplatedStructure*. *GeneratedStructure* copies keywords from the original query and adds question tokens to generate a new query without adding new concepts to the original query. *TemplatedStructure* contains all the keywords of the original query in the new query and lets the structure expand until it reaches a maximum threshold.
- (2) Expansion with concepts adds new concepts to the original query by using additional terms. *ClassicQEConcepts* combines the relevance model RM3, which involves the classic pseudo-relevance feedback. A method of automatic local analysis is provided to improve retrieval performance without additional interaction to discover the relevant concepts.
- (3) The *Hybrid* structure and concept expansions were implemented in a method called *GoogleQuestions*.

Figure 11 shows a comparison of these methods of query expansion. The experimental evaluation of these three methods [96] on Robust04 showed that *ClassicQEConcepts* based on discrete terms to expand the query delivered the worst performance, while *FilteredGoogleQuestions* based on structures and concepts delivered better performance on “Query Title” and “Query description.”

Because BERT can obtain contextualized embeddings, researchers have explored the advantages of query expansion. Zheng et al. proposed the BERT-QE model in Reference [156]. It is a method

Title query	Human genetic code
Description query	What progress is being made in the effort to map and sequence the human genetic code
GeneratedStructure	What is the human genetic code
TemplateStructure	What is the human genetic code?
ClassicQEConcepts	Gene dna genome genes research
GoogleQuestions	How long is the human genetic code? How many genes are in the human genome 2018? Who broke the DNA genetic code? What is human code?

Fig. 11. A comparison of methods of query expansion [96].

of query expansion based on BERT for document reranking that uses important information from **pseudo-relevance feedback (PRF)**. The BERT-QE model consists of three phases: (1) reranking the initial candidate documents retrieved by BM25 via BERT, (2) selecting fixed-length chunks for query expansion from the top-ranking documents obtained from phase one, and (3) reranking the documents with a combination of the original query and selected chunks. More details of the last two phases are shown in Figure 12. The results of experiments on the Robust04 and GOV2 datasets showed that BERT-QE outperformed BERT_{Large}. Moreover, BERT-QE-LLL (BERT-Large during three phases) recorded gains of 2.5%, 2.5%, and 3.3% on P@20, NDCG@20, and MAP@1k compared with BERT_{Large}, respectively. However, it had a 30% higher computational cost than BERT_{Large}. Moreover, all the above approaches failed to identify any new relevant document from the collection that had not been retrieved in the initial ranking.

Naseri et al. proposed **contextualized embeddings for query expansion (CEQE)** [113], which is similar to BERT-QE. It provides improvements based on a probabilistic language model combined with the PRF. It uses BERT to extract contextualized embeddings for the query as well as for the terms in the top-ranking feedback documents and then selects as expansion terms ones that are the closest to the query embeddings according to some similarity measure. In Robust04, the variations in the CEQE can yield improvements of 10.5% and 19.9% on P@20 and MAP@1000 over the BM25, respectively. Unlike BERT-QE, CEQE can be used to generate a new query of terms for execution on a classical inverted index, but this comes at the cost of losing the contextual meaning of an expansion term (e.g., adding a polysemous word to the query can lead to topic drift).

To address this issue, recent state-of-the-art approaches [35, 133] have exploited the advantages of using the PRF mechanism to improve query representations in scenarios of dense retrieval. The ColBERT-PRF [133] approach was the first to use PRF in a dense retrieval setting. As a robust mechanism of dense retrieval (a brief summary of ColBERT is provided in Section 3.4.1), ColBERT-PRF improves performance in terms of dense retrieval for query expansion in conjunction with PRF, which implements contextualized query expansion with dense retrieval, based on the pseudo-relevance set of documents identified from the first-pass retrieval. ColBERT-PRF utilizes k -means clustering to extract representative feedback embeddings while ensuring that they can be used to discriminate among the representative embeddings. Subsequently, the representative embeddings are incorporated into the query representation. Extensive experimental evaluation of ColBERT-PRF on the TREC 2019 and 2020 supports their claims as it significantly outperforms the relevant approaches.

Following the same pipeline, Yu et al. proposed ANCE-PRF [35], which improves query representations directly from the pseudo-relevance labels by utilizing the query and the top retrieved

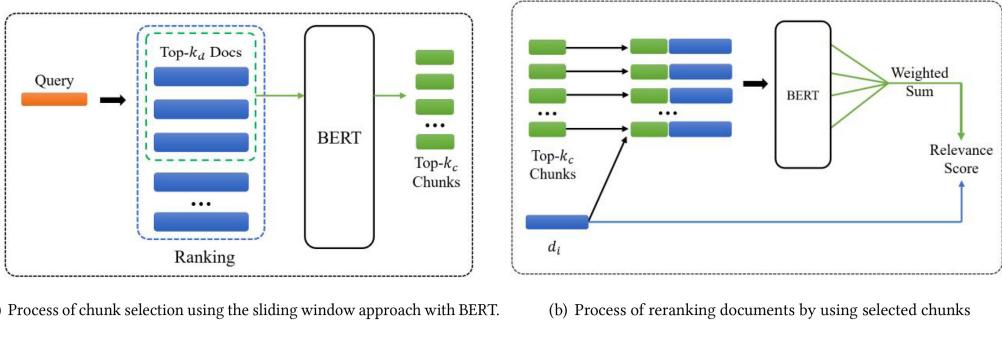


Fig. 12. Schematic illustration of BERT-QE proposed by Zheng et al. [156].

documents from a dense retrieval model, ANCE [69]. In this way, it learns to generate better query embeddings directly from the relevant labels such that the document index remains unchanged and this reduces overhead. The experimental analysis reported for ANCE-PRF [35] also indicated that under the same experimental settings, the margin of improvement obtained by using ANCE-PRF over its baseline ANCE was higher than that obtained by using strong models for dense retrievals, such as ColBERT or DistillBERT.

3.7 BERT-based Document Expansion

Inspired by methods of query expansion, researchers have explored expansions based on the documents themselves. Yan et al. proposed the *Doc2query* method to expand the document before indexing [100]. Given a document d , the *Doc2query* model predicts a query related to it. The expanded document is a concatenation of the original document and the predicted query. In contrast to other methods of query expansion that expand query representations, *Doc2query* aims to improve its capability of retrieval by expanding the representations of the documents. The retrieved candidate documents are fed into BERT for reranking. The results of experiments showed that BM25 yielded an improvement of more than 15% in terms of the MAP and MRR@10 on both the TREC-CAR and the MS MARCO datasets by using *Doc2query*. In the same context, Yan et al. [84] proposed a document expansion-based retriever that first generates a query set for each passage, based on the pointer-generator model [11], and then performs indexing. After the retrieval stage, the candidate documents are fed into BERT for reranking. Inspired by *StructBERT* [130], which incorporates the structural objectives of the word and the sentence to reconstruct their correct orders, the authors of this article addressed the challenging task of predicting the sentences preceding and following a given sentence, instead of the original NSP task. The results of comprehensive experimental evaluations in Reference [84] verified its superiority over the relevant state-of-the-art approaches in terms of both passage ranking and document ranking.

4 RESOURCES FOR BERT-BASED RANKING MODELS

In this section, we first present different datasets used in ad hoc IR applications and then detail the resources available for BERT for IR, including MatchZoo [45], Anserini [93], Pyserini [49], Birch [143], PyTerrier [22], Capreolus [14], OpenNIR [109], and OpenMatch [151], in Section 4.2.

4.1 Datasets

Table 4 provides information on the datasets that are typically used in ad hoc IR applications. We describe them below.

Table 4. Statistical Information on Collections

Collection	Query Creation	# Queries	# Documents
MS MARCO (Dev)	Manual	6,980	8.8M
MS MARCO (Eval)	Manual	6,837	8.8M
MS MARCO v2 (Dev1)	Manual	3,903	138M
MS MARCO v2 (Dev2)	Manual	4,281	138M
TREC CAR	Automatic	3.7M	29M
Robust04	Manual	250	0.5M
ClueWeb09-B	Automatic/Manual	200	50M
ClueWeb12-B13	Automatic/Manual	100	52M
TREC 2011	Automatic/Manual	50	16M
TREC 2012	Automatic/Manual	60	16M
TREC 2013	Automatic/Manual	60	240M
TREC 2014	Automatic/Manual	55	240M

MS MARCO: This dataset [122] consists of 8.8M passages for the ranking task. The training set contains 500K pairs of queries and the relevant documents, and the development set contains 6,980 queries and an evaluation set with approximately 6,800 queries. In July 2021, Microsoft released an updated version of the MS MARCO collections called **Version 2 (V2)**. Unlike the first version, in which the documents had been collected at different points in time, the passages and documents in MS MARCO V2 were created from the same underlying Bing web crawl. All the comparative statistics for MS MARCO V2 are shown in Table 4.

TREC CAR: The TREC CAR dataset [66] consists of 29M documents extracted from English Wikipedia and has five folds. The first four folds contain approximately 3M questions and the last fold contains 700K questions.

Robust04: TREC 2004 Robust Track is a standard ad hoc retrieval dataset [31] containing 250 topics over a newswire corpus of approximately 500K documents.

ClueWeb09-B⁴: ClueWeb09-B consists of 200 queries (i.e., Topics 1-50, 51-100, 101-150, 151-200) with relevance labels from TREC Web Track 2009-2012 and contains 50M documents.

ClueWeb12-B13: ClueWeb12-B13 contains 100 queries from TREC Web Track 2013-2014 and 52M webpages.

TREC Microblog: This consists of test tweet collections from the TREC Microblog Tracks 2011 to 2014 [52]. The salient targets of the track are social networks and real-time search. The TREC 2011 Microblog Tracks contains 50 queries and approximately 16 million tweets sampled from January 23 to February 8, 2011, from Twitter. For TREC 2012, the organizers used a new set of 60 topics and the same corpus as that of TREC 2011. TREC 2013 consists of 60 queries and approximately 240M tweets collected from Twitter from February 1 to March 31, 2013. TREC 2014 contains a set of 55 queries and the same collection as TREC 2013.

4.2 Resources for BERT-based Ranking Model

*MatchZoo*⁵ is a text-matching toolkit that focuses on developing text-matching models. This includes the design and comparison of models. Examples of deep matching models include the DRMM, MatchPyramid [68], MV-LSTM [117], DUET [20], ARC-I, ARC-II, and DSSM. MatchZoo uses a unified interface design for tasks including document retrieval, question-answering, session response ranking, and synonym recognition. It contains three main modules for data

⁴<https://github.com/cdegroc/warc-clueweb>

⁵<https://github.com/NTMC-Community/MatchZoo-py>

preprocessing, model building, and training and evaluation. Developed based on Keras, it supports TensorFlow, the Computational Network Toolkit, and Theano and works seamlessly on both CPUs and GPUs. MatchZoo focuses on the implementation of prevalent neural ranking models.

*Anserini*⁶ provides wrappers and extensions based on core Lucene libraries to perform common research tasks by using intuitive APIs. The results of experiments on ad hoc document retrieval have shown that BM25 (Anserini) is superior to BM25, because it performs better in first-stage retrieval.

*Pyserini*⁷ is an easy-to-use Python toolkit for research on reproducible information retrieval with sparse and dense representations. The aim is to support the IR research community by making use of effective, reproducible, and easy-to-use first-stage retrieval in a multi-stage ranking architecture.

*Birch*⁸ is a framework for document ranking via BERT. It was written by using a combination of Python and Java Virtual Machine code. The top k candidate documents retrieved by Anserini are fed into BERT for reranking.

*PyTerrier*⁹ is a Python framework for conducting experiments on information retrieval based on Terrier,¹⁰ which provides an interface for downloading datasets and experimental functions. This allows for adequate comparisons between approaches to retrieval on the same queries and evaluation metrics.

*Capreolus*¹¹ is an NLP toolkit for ad hoc document retrieval to enable mutual conversion between modules.

*OpenNIR*¹² is a library for training and evaluating ranking tasks. It provides important information for fair comparisons, including a reranking threshold and the loss function. It is implemented on PyTerrier.

*OpenMatch*¹³ is an open source package for IR that uses neural ranking models for matching and understanding deep text. It can be used for document retrieval, document reranking, and transfer learning across domains.

PyTerrier can provide interfaces for a detailed experimental setup to save the time needed to implement state-of-the-art indexing and retrieval functionalities. This enables the rapid development and evaluation of large-scale retrieval applications. Capreolus, OpenNIR, and OpenMatch are more suitable toolkits for drawing fair comparisons among state-of-the-art IR approaches by tuning the hyperparameters during training.

5 CONCLUSIONS AND FUTURE WORK

In this article, we provided a survey of the application of the BERT model to information retrieval. Because the BERT model is based on the pretrained transformer encoder architecture, we also studied other pretrained transformer encoders that are based on the BERT architecture. In our review, we summarized advanced methods based on the BERT architecture to deal with long documents, integrate semantic information, balance effectiveness and efficiency, predict term weights, and expand queries and documents. Furthermore, we analyzed the differences between approaches based on the BERT model to enable the reader to clearly understand the advantages and disadvantages

⁶<http://anserini.io/>

⁷<https://github.com/castorini/pyserini>

⁸<https://github.com/castorini/birch>

⁹<https://github.com/fennuDetudou/tudouNLP>

¹⁰<http://terrier.org/>

¹¹<https://github.com/capreolus-ir/capreolus>

¹²<https://github.com/Georgetown-IR-Lab/OpenNIR>

¹³<https://github.com/thunlp/OpenMatch>

of each. Due to inconsistencies in the datasets and evaluation metrics used for different models, we compared their published experimental results on benchmark datasets.

We presented resources for utilizing BERT for IR, including MatchZoo [45], Anserini [93], Pyserini [49], Birch [143], PyTerrier [22], Capreolus [14], OpenNIR [109], and OpenMatch [151]. We also discussed outstanding challenges to the use of BERT for IR according to two perspectives: (1) methods of aggregating long documents and (2) approaches to balance effectiveness and efficiency. In addition, we emphasize the benefits of using encoder-based BERT models over recent LLMs such as ChatGPT, which are decoder-based and require significant computational power.

The experimental evaluations of state-of-the-art methods showed that their accuracy on different datasets in terms of MRR@10 was around 50%. Hence, there is considerable room for improvement. A number of directions can be adopted in future research to this end. These include a novel combinatorial function to merge the contextualized embeddings of the BERT model with other neural ranking models for neural IR via multi-task optimization [1, 75], and reinforcement learning techniques [36, 72] to transform weakly supervised learning into unsupervised learning to reduce the cost of training the data. Powerful generative large language models [4, 38, 39, 80, 87, 95] can be leveraged across various stages of retrieval and ranking processes [129, 152] and be applied for more applications (such as biomedicine and chemical IR) in the future [55, 76, 77, 134]. Moreover, these models can be beneficial in enhancing techniques of query and document expansion [67]. Exploring the integration of Conversational Search Systems [64] with multimodal IR presents a promising future direction for more dynamic, user-centric interactions and precise information retrieval through chat and voice technologies.

ACKNOWLEDGMENTS

We gratefully appreciate all the anonymous reviewers and the associate editor for their valuable comments and constructive suggestions that greatly helped improve the quality of the article. The authors thank Prof. Xinhua Tu for his fruitful discussion when he visited our lab at York University. The authors also thank Dr. Junmei Wang for her contribution at the early stage of research when she was a Ph.D. candidate at Central China Normal University. All the work was done when the first author was a Ph.D. candidate.

REFERENCES

- [1] A. Abolghasemi, S. Verberne, and L. Azzopardi. 2022. Improving BERT-based query-by-document retrieval with multi-task optimization. In *European Conference on Information Retrieval*. Springer, Cham.
- [2] A. Babashzadeh, M. Daoud, and J. X. Huang. 2023. Using semantic-based association rule mining for improving clinical text retrieval. In *Health Information Science: Second International Conference, HIS 2013*. Springer, Berlin, 2013.
- [3] A. Baddeley. 1992. Working memory. *Science* 255, 5044 (1992), 556–559.
- [4] A. Chowdhery, S. Narang, J. Devlin, M. Bosma et al. 2023. Palm: Scaling language modeling with pathways. *J. Mach. Learn. Res.* 24, 240 (2023), 1–113.
- [5] A. Graves and J. Schmidhuber. 2005. Framewise phoneme classification with bidirectional lstm and other neural network architectures. *Neural Netw.* 18, 5–6 (2005), 602–610.
- [6] A. H. Lashkari, F. Mahdavi, and V. Ghomi. 2009. A boolean model in information retrieval for search engines. In *Proceedings of the International Conference on Information Management and Engineering*. IEEE, 385–389.
- [7] A. Khan, A. Sohail, U. Zahoor, and A. S. Qureshi. 2020. A survey of the recent architectures of deep convolutional neural networks. *Artif. Intell. Rev.* 53, 8 (2020), 5455–5516. <https://doi.org/10.1007/s10462-020-09825-6>
- [8] A. Krizhevsky, I. Sutskever, and G. E. Hinton. 2017. Imagenet classification with deep convolutional neural networks. *Commun. ACM* 60, 6 (2017), 84–90.
- [9] A. Mallia, O. Khattab, T. Suel, and N. Tonello. 2021. Learning passage impacts for inverted indexes. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1723–1727.
- [10] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever. 2018. Improving language understanding by generative pre-training.

- [11] A. See, P. J. Liu, and C. D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*. 1073–1083.
- [12] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. 2017. Attention is all you need. In *Proceedings of Advances in Neural Information Processing Systems*, 30.
- [13] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, and S. R. Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. arXiv:1804.07461. Retrieved from <https://arxiv.org/abs/1804.07461>
- [14] A. Yates, S. Arora, X. Zhang, W. Yang, K. M. Jose, and J. Lin. 2020. Capreolus: A toolkit for end-to-end neural ad hoc retrieval. In *Proceedings of the 13th International Conference on Web Search and Data Mining*. 861–864.
- [15] A. Yadav and D. K. Vishwakarma. 2020. Sentiment analysis using deep learning architectures: A review. *Artif. Intell. Rev.* 53, 6 (2020), 4335–4385.
- [16] B. He, J. X. Huang, and X. Zhou. 2011. Modeling term proximity for probabilistic information retrieval models. *Inf. Sci.* 181, 14 (2011), 3017–3031.
- [17] B. Hu, Z. Lu, H. Li, and Q. Chen. 2014. Convolutional neural network architectures for matching natural language sentences. In *Proceedings of the 27th International Conference on Neural Information Processing Systems, Volume 2*. 2042–2050.
- [18] C. Xiong, Z. Dai, and J. Callan. 2017. 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*. 55–64.
- [19] B. McCann, J. Bradbury, C. Xiong, and R. Socher. 2017. Learned in translation: Contextualized word vectors. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*. 6297–6308.
- [20] B. Mitra, F. Diaz, and N. Craswell. 2017. Learning to match using local and distributed representations of text for web search. In *Proceedings of the 26th International Conference on World Wide Web*. 1291–1299.
- [21] B. Mitra and N. Craswell. 2017. Neural models for information retrieval. arXiv:1705.01509. Retrieved from <https://arxiv.org/abs/1705.01509>
- [22] C. Macdonald and N. Tonello. 2020. Declarative experimentation in information retrieval using PyTerrier. In *Proceedings of the ACM SIGIR on International Conference on Theory of Information Retrieval*. 161–168.
- [23] C. Li, A. Yates, S. MacAvaney, B. He, and Y. Sun. 2020. PARADE: Passage representation aggregation for document reranking. arXiv:2008.09093. Retrieved from <https://arxiv.org/abs/2008.09093>
- [24] C. Luo, Y. Zheng, J. Mao, Y. Liu, M. Zhang, and S. Ma. 2017. Training deep ranking model with weak relevance labels. In *Australasian Database Conference*. Springer, Cham, 205–216.
- [25] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. arXiv:1910.10683. Retrieved from <https://arxiv.org/abs/1910.10683>
- [26] D. Cer, Y. Yang, S. Kong, N. Hua, N. Limtiaco, R. S John, N. Constant, M. Guajardo-Cespedes, S. Yuan, C. Tar, Y.-H. Sung, B. Strope, and R. Kurzweil. 2018. Universal sentence encoder. arXiv: 1803.11175. Retrieved from <https://arxiv.org/abs/1803.11175>
- [27] D. Hiemstra. 2001. Using language models for information retrieval. *Taaluitgeverij Neslia Paniculata*, CTIT Ph.D. Thesis Series No. 01-32.
- [28] D. W. Otter, J. R. Medina, and J. K. Kalita. 2020. A survey of the usages of deep learning for natural language processing. *IEEE Trans. Neural Netw. Learn. Syst.* 32, 2 (2020), 604–624.
- [29] E. Alsentzer, J. R. Murphy, W. Boag, W. H. Weng, D. Jin, T. Naumann, and M. McDermott. 2019. Publicly available clinical BERT embeddings. arXiv:1904.03323. Retrieved from <https://arxiv.org/abs/1904.03323>
- [30] E. Kamalloo, X. Zhang, O. Ogundepo, N. Thakur, D. Alfonso-Hermelo, M. Rezagholizadeh, and J. Lin. 2023. Evaluating embedding APIs for information retrieval. arXiv:2305.06300. Retrieved from <https://arxiv.org/abs/2305.06300>
- [31] E. M. Voorhees. 2004. Overview of the TREC 2004 robust track. In *Proceedings of the 23rd Text REtrieval Conference (TREC '04)*. 52–69.
- [32] G. Hinton, L. Deng, D. Yu, G. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, B. Kingsbury, and T. Sainath. 2012. Deep neural networks for acoustic modeling in speech recognition. *IEEE Sign. Process. Mag.* 29, 6 (2012), 82–97.
- [33] G. Salton, A. Wong, and C. S. A. Yang. 1975. A vector space model for automatic indexing. *Commun. ACM* 18, 11 (1975), 613–620.
- [34] G. Salton, E. A. Fox, and H. Wu. 1983. Extended boolean information retrieval. *Commun. ACM* 26, 11 (1983) 1022–36.
- [35] H. C. Yu, C. Xiong, and J. Callan. 2021. Improving query representations for dense retrieval with pseudo-relevance feedback. arXiv:2108.13454. Retrieved from <https://arxiv.org/abs/2108.13454>
- [36] H. Gharagozlu, J. Mohammadzadeh, A. Bastanfar, and S. S. Ghidary. 2022. RLAS-BIABC: A reinforcement learning-based answer selection using the BERT model boosted by an improved ABC algorithm. *Comput. Intell. Neurosci.* (2022).

- [37] H. Schütze, C. D. Manning, and P. Raghavan. 2008. *Introduction to Information Retrieval*. Cambridge University Press, Cambridge, UK.
- [38] H. Touvron, T. Lavigra, G. Izacard, X. Martinet et al. 2023. Llama: Open and efficient foundation language models. arXiv:2302.13971. Retrieved from <https://arxiv.org/abs/2302.13971>
- [39] H. Touvron, L. Martin, K. Stone, P. Albert et al. 2023. Llama 2: Open foundation and fine-tuned chat models. arXiv:2307.09288. Retrieved from <https://arxiv.org/abs/2307.09288>
- [40] H. Zamani, and W. B. Croft. 2018. On the theory of weak supervision for information retrieval. In *Proceedings of the ACM SIGIR International Conference on Theory of Information Retrieval*. 147–154.
- [41] H. Zamani, W. B. Croft, and J. S. Culpepper. 2018. Neural query performance prediction using weak supervision from multiple signals. In *Proceedings of the 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. 105–114.
- [42] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. 2020. Generative adversarial nets. *Commun. ACM* 63, 11 (2020), 139–144.
- [43] I. Jahan, M. T. R. Laskar, C. Peng, J. X. Huang. 2022. Evaluation of ChatGPT on biomedical tasks: A zero-shot comparison with fine-tuned generative transformers. In *Proceedings of the 22nd Workshop on Biomedical Natural Language Processing and BioNLP Shared Tasks*. 326–336.
- [44] J. Devlin, M. W Chang, K. Lee, and K. Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 4171–4186.
- [45] J. Guo, Y. Fan, X. Ji, and X. Cheng. 2019. Matchzoo: A learning, practicing, and developing system for neural text matching. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1297–1300.
- [46] J. Guo, Y. Fan, Q. Ai, and W. B. Croft. 2016. A deep relevance matching model for ad-hoc retrieval. In *Proceedings of the 25th ACM International Conference on Information and Knowledge Management*. 55–64.
- [47] J. Guo, Y. Fan, L. Pang, L. Yang, Q. Ai, H. Zamani, C. Wu, W. B. Croft, and X. Cheng. 2020. A deep look into neural ranking models for information retrieval. *Inf. Process. Manage.* 57, 6 (2020), 102067.
- [48] J. Lee, W. Yoon, S. Kim, D. Kim, S. Kim, C. H. So, and J. Kang. 2020. BioBERT: A pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics* 36, 4 (2020), 1234–1240.
- [49] J. Lin, X. Ma, S. C. Lin, J. H. Yang, R. Pradeep, and R. Nogueira. 2021. Pyserini: An easy-to-use python toolkit to support replicable ir research with sparse and dense representations. arXiv:2102.10073. Retrieved from <https://arxiv.org/abs/2102.10073>
- [50] J. Lin, R. Nogueira, and A. Yates. 2021. Pretrained transformers for text ranking: BERT and beyond. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*.
- [51] J. Pennington, R. Socher, and C. D. Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP '14)*.
- [52] J. Lin, M. Efron, Y. Wang, and G. Sherman. 2014. Overview of the TREC-2014 microblog track. In *Proceedings of the 23rd Text REtrieval Conference (TREC '14)*.
- [53] J. Ponte and W. B. Croft. 1998. A language modeling approach to information retrieval. In *Proceedings of the 21st Annual International ACM SIGIR Conference* (1998), 275–281.
- [54] J. X. Huang, F. Peng, D. Schuurmans, and N. Cercone. 2003. Applying machine learning to text segmentation for information retrieval. *Inf. Retriev.* 6, 3 (2003), 333–362.
- [55] J. X. Huang and Q. Hu. 2009. A bayesian learning approach to promoting diversity in ranking for biomedical information retrieval. In *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval*.
- [56] J. X. Huang, M. Zhong, and L. Si. 2005. York University at TREC 2005: Genomics Track. In *TREC*. 2005.
- [57] J. X. Huang, J. Miao, and B. He. 2013. High performance query expansion using adaptive co-training. *Inf. Process. Manage.* 49, 2 (2013), 441–453.
- [58] J. Zhan, J. Mao, Y. Liu, M. Zhang, and S. Ma. 2020. RepBERT: Contextualized text embeddings for first-stage retrieval. 2020. arXiv:2006.15498. Retrieved from <https://arxiv.org/abs/2006.15498>
- [59] J. Zhao, J. X. Huang, and Z. Ye. 2014. Modeling term associations for probabilistic information retrieval. *ACM Trans. Inf. Syst.* 32, 2 (2014), 1–47.
- [60] J. Zhao, J. X. Huang, and B. He. 2011. CRTER: Using cross terms to enhance probabilistic information retrieval. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*.
- [61] K. D. Onal, Y. Zhang, I. S. Altingovde, M. M. Rahman, P. Karagoz, A. Braylan, B. Dang, H. L. Chang, H. Kim, Q. McNamara, and A. Angert. 2018. Neural information retrieval: At the end of the early years. *Inf. Retriev.* 21, 2 (2018) 111–182.

- [62] K. Clark, M. T. Luong, Q. V. Le, and C. D. Manning. 2020. ELECTRA: Pre-training text encoders as discriminators rather than generators. In *Proceedings of International Conference on Learning Representations (ICLR '20)*.
- [63] K. Hui, A. Yates, K. Berberich, and G. Melo. 2018. Co-PACRR: A context-aware neural IR model for ad-hoc retrieval. In *Proceedings of the 11th ACM International Conference on Web Search and Data Mining*. 279–287.
- [64] K. Keyvan and J. X. Huang. 2022. How to approach ambiguous queries in conversational search: A survey of techniques, approaches, tools, and challenges. *ACM Comput. Surv.* 55, 6 (2022), 1–40.
- [65] K. Zhang, C. Xiong, Z. Liu, and Z. Liu. 2020. Selective weak supervision for neural information retrieval. In *Proceedings of The Web Conference*. 474–485.
- [66] L. Dietz, M. Verma, F. Radlinski, and N. Craswell. 2017. TREC complex answer retrieval overview. In *Proceedings of the Text REtrieval Conference (TREC '17)*.
- [67] L. Liu, M. Li, J. Lin, S. Riedel, and P. Stenetorp. 2022. Query expansion Using contextual clue sampling with language models. arXiv:2210.07093. Retrieved from <https://arxiv.org/abs/2210.07093>
- [68] L. Pang, Y. Lan, J. Guo, J. Xu, S. Wan, and X. Cheng. 2016. Text matching as image recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*. 2793–2799.
- [69] L. Xiong, C. Xiong, Y. Li, K. Tang, and J. Liu. 2020. Approximate nearest neighbor negative contrastive learning for dense text retrieval. arXiv:2007.00080. Retrieved from <https://arxiv.org/abs/2007.00080>
- [70] M. Dehghani, A. Severyn, S. Rothe, and J. Kamps. 2017. Learning to learn from weak supervision by full supervision. arXiv:1711.11383. Retrieved from <https://arxiv.org/abs/1711.11383>
- [71] M. Ding, C. Zhou, H. Yang, and J. Tang. 2020. Cogltx: Applying bert to long texts. In *Advances in Neural Information Processing Systems*, Vol. 33, 12792–12804.
- [72] M. Gaur, K. Gunaratna, V. Srinivasan, and H. Jin. 2022. Isseq: Information seeking question generation using dynamic meta-information retrieval and knowledge graphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*. 10672–10680.
- [73] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer. 2018. Deep contextualized word representation. In *Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 2227–2237.
- [74] M. Li and E. Gaussier. 2021. KeyBLD: Selecting key blocks with local pre-ranking for long document information retrieval. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2207–2211.
- [75] M. Li and E. Gaussier. 2022. BERT-based dense intra-ranking and contextualized late interaction via multi-task learning for long document retrieval. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*.
- [76] M. Lupu, J. X. Huang, J. Zhu, and J. Tait. 2009. TREC-CHEM: Large scale chemical information retrieval evaluation at TREC. In *Proceedings of the ACM SIGIR Forum*. 63–70.
- [77] M. Lupu, J. X. Huang, J. Zhu, and J. Tait. 2009. Overview of the TREC 2009 chemical IR track. In *Proceedings of the Text REtrieval Conference (TREC '09)*.
- [78] M. Pan, J. Wang, J. X. Huang, A. J. Huang, Q. Chen, and J. Chen. 2022. A probabilistic framework for integrating sentence-level semantics via BERT into pseudo-relevance feedback. *Inf. Process. Manage.* 59, 1 (2022), 102734.
- [79] M. Pan, Q. Pei, Y. Liu, T. Li, E. A. Huang, J. Wang, and J. X. Huang. 2023. SPRF: A semantic pseudo-relevance feedback enhancement for information retrieval via ConceptNet. *Knowl.-Bas. Syst.* 274, 110602 (2023).
- [80] M. T. R. Laskar, M. S. Bari, M. Rahman, M. A. H. Bhuiyan, S. Joty, and J. X. Huang. 2023. A systematic study and comprehensive evaluation of ChatGPT on benchmark datasets. In *Findings of the Association for Computational Linguistics*. 431–469.
- [81] M. T. R. Laskar, J. X. Huang, and E. Hoque. 2020. Contextualized embeddings-based transformer encoder for sentence similarity modeling in answer selection task. In *Proceedings of the 12th Language Resources and Evaluation Conference*.
- [82] M. T. R. Laskar, E. Hoque, and J. X. Huang. 2022. Domain adaptation with pre-trained transformers for query-focused abstractive text summarization. *Comput. Ling.* 48, 2 (2022), 279–320.
- [83] M. T. R. Laskar, E. Hoque, and J. X. Huang. 2020. WSL-DS: Weakly supervised learning with distant supervision for query-focused multi-document abstractive summarization. In *Proceedings of the 28th International Conference on Computational Linguistics*. 5647–5654.
- [84] M. Yan, C. Li, C. Wu, B. Bi, W. Wang, J. Xia, and L. Si. 2019. IDST at TREC 2019 deep learning track: Deep cascade ranking with generation-based document expansion and pre-trained language modeling. In *Proceedings of the Text REtrieval Conference (TREC '19)*.
- [85] N. Reimers and I. Gurevych. 2019. Sentence-BERT: Sentence embeddings using siamese BERT-networks. arXiv:1908.10084. Retrieved from <https://arxiv.org/abs/1908.10084>
- [86] O. Khattab and M. Zaharia. 2020. 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*. 39–48.

- [87] R. OpenAI. 2023. GPT-4 technical report. arXiv:2303.08774. View in Article, 2, 13.
- [88] P. Lewis, E. Perez, A. Piktus, F. Petroni et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Advances in Neural Information Processing Systems*, Vol. 33, 9459–9474.
- [89] P. S. Huang, X. He, J. Gao, L. Deng, A. Acero, and L. Heck. 2013. Learning deep structured semantic models for web search using clickthrough data. In *Proceedings of the 22nd ACM International Conference on Information & Knowledge Management*. 2333–2338.
- [90] P. Shi and J. Lin. 2019. Cross-lingual relevance transfer for document retrieval. arXiv:1911.02989. Retrieved from <https://arxiv.org/abs/1911.02989>
- [91] P. Punyani, R. Gupta, and A. Kumar. 2020. Neural networks for facial age estimation: A survey on recent advances. *Artif. Intell. Rev.* 53, 5 (2020), 3299–3347.
- [92] P. Xu, X. Ma, R. Nallapati, and B. Xiang. 2019. Passage ranking with weak supervision. arXiv:1905.05910. Retrieved from <https://arxiv.org/abs/1905.05910>
- [93] P. Yang, H. Fang, and J. Lin. 2017. Anserini: Enabling the use of lucene for information retrieval research. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1253–1256.
- [94] Q. Chen, Q. Hu, J. X. Huang, L. He, and W. An. 2017. Enhancing recurrent neural networks with positional attention for question answering. 2017. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*.
- [95] R. Anil, A. M. Dai, O. Firat, M. Johnson, D. Lepikhin et al. 2023. Palm 2 technical report. arXiv:2305.10403. Retrieved from <https://arxiv.org/abs/2305.10403>
- [96] R. Padaki, Z. Dai, and J. Callan. 2020. Rethinking query expansion for BERT reranking. In *European Conference on Information Retrieval*. Springer, Cham, 297–304.
- [97] R. Nogueira, J. Lin, and A. I. Epistemic. 2019. From doc2query to docTTTTquery. Online preprint, 6, 2.
- [98] R. Nogueira and K. Cho. 2019. Passage Re-ranking with BERT. arXiv:1901.04085. Retrieved from <https://arxiv.org/abs/1901.04085>
- [99] R. Nogueira, W. Yang, K. Cho, and J. Lin. 2019. Multi-stage document ranking with BERT. arXiv:1904.08375. Retrieved from <https://arxiv.org/abs/1904.08375>
- [100] R. Nogueira, W. Yang, J. Lin, and K. Cho. 2019. Document expansion by query prediction. arXiv:1904.08375. Retrieved from <https://arxiv.org/abs/1904.08375>
- [101] R. Sennrich, H. Barry, and B. Alexandra. 2015. Neural machine translation of rare words with subword units. arXiv:1508.07909. Retrieved from <https://arxiv.org/abs/1508.07909>
- [102] R. Zhu, X. Tu, and J. X. Huang. 2020. Deep learning on information retrieval and its applications. In *Deep Learning for Data Analytics*. Academic Press, San Diego, CA, 125–153.
- [103] S. C. Lin, J. H. Yang, and J. Lin. 2020. Distilling dense representations for ranking using tightly-coupled teachers. arXiv:2010.11386. Retrieved from <https://arxiv.org/abs/2010.11386>
- [104] S. Hofstätter, M. Zlabinger, and A. Hanbury. 2019. TU Wien TREC deep learning'19—Simple contextualization for re-ranking. arXiv:1912.01385. Retrieved from <https://arxiv.org/abs/1912.01385>
- [105] S. Hofstätter, M. Zlabinger, and A. Hanbury. 2020. Interpretable & time-budget-constrained contextualization for re-ranking. arXiv:2002.01854. Retrieved from <https://arxiv.org/abs/2002.01854>
- [106] S. Hofstätter, H. Zamani, B. Mitra, N. Craswell, and A. Hanbury. 2020. Local self-attention over long text for efficient document retrieval. In *Proceedings of the 43rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '20)*. 2021–2024.
- [107] S. Hofstätter, S. C. Lin, J. H. Yang, J. Lin, and A. Hanbury. 2021. Efficiently teaching an effective dense retriever with balanced topic aware sampling. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 113–122.
- [108] S. Hofstätter, B. Mitra, H. Zamani, N. Craswell, and A. Hanbury. 2021. Intra-document cascading: Learning to select passages for neural document ranking. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1349–1358.
- [109] S. MacAvaney. 2020. OpenNIR: A complete neural ad-hoc ranking pipeline. In *Proceedings of the 13th International Conference on Web Search and Data Mining*. 845–848.
- [110] S. MacAvaney, F. M. Nardini, R. Perego, N. Tonello, N. Goharian, and O. Frieder. 2020. Expansion via prediction of importance with contextualization. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1573–1576.
- [111] S. MacAvaney, A. Yates, A. Cohan, and N. Goharian. 2019. CEDR: Contextualized embeddings for document ranking. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1101–1104. <https://doi.org/10.1145/3331184.3331317>
- [112] S. Macavaney, F. M. Nardini, R. Perego, N. Tonello, N. Goharian, and O. Frieder. 2020. Training curricula for open domain answer re-ranking. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 529–538.

- [113] S. Naseri, J. Dalton, A. Yates, and J. Allan. 2021. CEQE: Contextualized embeddings for query expansion. In *European Conference on Information Retrieval*. Springer, Cham, 467–482.
- [114] S. Robertson and H. Zaragoza. 2009. The probabilistic relevance framework: BM25 and beyond. *Found. Trends Inf. Retriev.* 3, 4 (2009), 333–389.
- [115] S. Wang, S. Zhuang, and G. Zuccon. 2021. BERT-based dense retrievers require interpolation with BM25 for effective passage retrieval. In *Proceedings of the ACM SIGIR International Conference on Theory of Information Retrieval*. 317–324. DOI: <https://doi.org/10.1145/3471158.3472233>
- [116] S. Wan, Y. Lan, J. Xu, J. Guo, L. Pang, and X. Cheng. 2016. Match-SRNN: modeling the recursive matching structure with spatial RNN. In *Proceedings of the 25th International Joint Conference on Artificial Intelligence*. 2922–2928.
- [117] S. Wan, Y. Lan, J. Guo, J. Xu, L. Pang, and X. Cheng. 2016. A deep architecture for semantic matching with multiple positional sentence representations. In *Proceedings of the AAAI Conference on Artificial Intelligence*.
- [118] S. Zhang, L. Yao, A. Sun, and Y. Tay. 2019. Deep learning based recommender system: A survey and new perspectives. *ACM Comput. Surv.* 52, 1 (2019), 1–38.
- [119] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam et al. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, Vol. 33, 1877–1901.
- [120] T. Formal, B. Piwowarski, and S. Clinchant. 2021. SPLADE: Sparse lexical and expansion model for first stage ranking. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2288–2292.
- [121] T. Mikolov, K. Chen, G. Corrado, and J. Dean. 2013. Efficient estimation of word representations in vector space. arXiv:1301.3781. Retrieved from <https://arxiv.org/abs/1301.3781>
- [122] T. Nguyen, M. Rosenberg, X. Song, J. Guo, S. Tiwary, R. Majumder, and L. Deng. 2016. MS MARCO: A human generated machine reading comprehension dataset. In *Proceedings of the Workshop on Cognitive Computation: Integrating Neural and Symbolic Approaches Co-located with the 30th Annual Conference on Neural Information Processing Systems (NIPS '16)*.
- [123] V. Karpukhin, B. Oğuz, S. Min, P. Lewis, L. Wu, S. Edunov, D. Chen, and W. Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP '20)*.
- [124] V. Sanh, L. Debut, J. Chaumond, and T. Wolf. 2019. DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter. arXiv:1910.01108. Retrieved from <https://arxiv.org/abs/1910.01108>
- [125] W. B. Croft, D. Metzler, and T. Strohman. 2010. *Search Engines: Information Retrieval in Practice*. Addison-Wesley, Reading, MA.
- [126] W. Liu, Z. Wang, X. Liu, N. Zeng, Y. Liu, and F. E. Alsaadi. 2017. A survey of deep neural network architectures and their applications. *Neurocomputing* 234 (2017), 11–26. DOI: <https://doi.org/10.1016/j.neucom.2016.12.038>
- [127] W. Liu, P. Zhou, Z. Zhao, Z. Wang, H. Deng, and Q. Ju. FastBERT: A self-distilling BERT with adaptive inference time. 2020. arXiv:2004.02178. Retrieved from <https://arxiv.org/abs/2004.02178>
- [128] W. Lu, J. Jiao, and R. Zhang. 2020. Twinbert: Distilling knowledge to twin-structured compressed bert models for large-scale retrieval. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 2645–2652.
- [129] W. Sun, L. Yan, X. Ma, P. Ren, D. Yin, and Z. Ren. 2023. Is ChatGPT good at search? Investigating large language models as re-ranking agent. arXiv:2304.09542. Retrieved from <https://arxiv.org/abs/2304.09542>
- [130] W. Wang, B. Bi, M. Yan, C. Wu, Z. Bao, J. Xia, L. Peng, and Si L. 2019. StructBERT: Incorporating language structures into pre-training for deep language understanding. arXiv: 1908.04577. Retrieved from <https://arxiv.org/abs/1908.04577>
- [131] W. Yang, H. Zhang, and J. Lin. 2019. Simple applications of BERT for ad hoc document retrieval. arXiv: 1903.10972. Retrieved from <https://arxiv.org/abs/1903.10972>
- [132] X. Jiao, Y. Yin, L. Shang, X. Jiang, X. Chen, L. Li, F. Wang, and Q. Liu. 2020. TinyBERT: Distilling BERT for natural language understanding. In *Findings of the Association for Computational Linguistics: EMNLP*. 4163–4174.
- [133] X. Wang, C. Macdonald, N. Tonello, and I. Ounis. 2021. Pseudo-relevance feedback for multiple representation dense retrieval. In *Proceedings of the ACM SIGIR International Conference on Theory of Information Retrieval*. 297–306.
- [134] X. Yin, J. X. Huang, Z. Li, and X. Zhou. 2013. A survival modeling approach to biomedical search result diversification using Wikipedia. *IEEE Trans. Knowl. Data Eng.* 25, 06 (2013), 1201–1212.
- [135] X. Zhang, A. Yates, and J. Lin. 2021. Comparing score aggregation approaches for document retrieval with pretrained transformers. In *European Conference on Information Retrieval*. Springer, Cham, 150–163.
- [136] Y. Bai, X. Li, G. Wang, C. Zhang, L. Shang, J. Xu, Z. Wang, F. Wang, and Q. Liu. 2020. SparTerm: Learning term-based sparse representation for fast text retrieval. arXiv:2010.00768. Retrieved from <https://arxiv.org/abs/2010.00768>
- [137] Y. Cui, W. Che, T. Liu, B. Qin, S. Wang, and G. Hu. 2020. Revisiting pre-trained models for Chinese natural language processing. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing: Findings*. 657–668.

- [138] Y. Lecun, Y. Bengio, and G. Hinton. 2015. Deep learning. *Nature* 521, 7553 (2015), 436–444. <https://doi.org/10.1038/nature14539>
- [139] Y. Liu, J. X. Huang, A. An, and X. Yu. 2007. ARSA: A sentiment-aware model for predicting sales performance using blogs. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*.
- [140] Y. Liu, M. Ott, N. Goyal, J. F. Du, M. Joshi, D. Q. Chen, O. Levy, M. Lewis, Zettlemoyer L, and Stoyanov V. Roberta. 2019. A robustly optimized BERT pretraining approach. arXiv:1907.11692. Retrieved from <https://arxiv.org/abs/1907.11692>
- [141] Y. Luan, J. Eisenstein, K. Toutanova, and M. Collins. 2021. Sparse, dense, and attentional representations for text retrieval. *Trans. Assoc. Comput. Ling.* 9 (2021), 329–345.
- [142] Y. Wu, M. Schuster, Z. Chen, Q. V.Le, and M. Norouzi. 2016. Google’s neural machine translation system: Bridging the gap between human and machine translation. arXiv:1609.08144. Retrieved from <https://arxiv.org/abs/1609.08144>
- [143] Z. A. Yilmaz, S. Wang, W. Yang, H. Zhang, and J. Lin. 2019. Applying BERT to document retrieval with birch. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP ’19)*, 19–24.
- [144] Z. A. Yilmaz, W. Yang, H. Zhang, and J. Lin. 2019. Cross-domain sentence modeling for relevance transfer with BERT. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP ’19)*.
- [145] Z. Dai, Z. Yang, Y. Yang, J. Carbonell, Q. Le, and R. Salakhutdinov. 2019. Transformer-XL: Attention language models beyond a fixed-length context. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2978–2988.
- [146] Z. Dai and J. Callan. 2019. 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*, 985–988.
- [147] Z. Dai and J. Callan. 2020. Context-aware passage term weighting for first stage retrieval. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1533–1536.
- [148] Z. Dai and J. Callan. 2020. Context-aware document term weighting for ad-hoc search. In *Proceedings of the Web Conference*, 1897–1907.
- [149] Z. Dai, C. Xiong, J. Callan, and Z. Liu. 2018. Convolutional neural networks for soft-matching n-grams in ad-hoc search. In *Proceedings of the 11th ACM International Conference on Web Search and Data Mining*, 126–134.
- [150] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut. 2019. ALBERT: A lite BERT for self-supervised learning of language representations. arXiv:1909.11942. Retrieved from <https://arxiv.org/abs/1909.11942>
- [151] Z. Liu, K. Zhang, C. Xiong, Z. Liu, and M. Sun. 2021. OpenMatch: An open source library for Neu-IR research. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2531–2535.
- [152] Z. Qin, R. Jagerman, K. Hui, H. Zhuang, J. Wu, J. Shen, T. Liu, J. Liu, D. Metzler, X. Wang et al. 2023. Large language models are effective text rankers with pairwise ranking prompting. arXiv:2306.17563. Retrieved from <https://arxiv.org/abs/2306.17563>
- [153] Z. Wu, J. Mao, Y. Liu, J. Zhan, Y. Zheng, M. Zhang, and S. Ma. 2020. Leveraging passage-level cumulative gain for document ranking. In *Proceedings of the Web Conference*, 2421–2431.
- [154] Z. Yang, Z. Dai, Y. Yang, and J. Carbonell. 2019. XLNET: Generalized autoregressive pretraining for language understanding. In *Proceedings of Advances in Neural Information Processing Systems*, Vol. 32, 5753–5763.
- [155] Z. Ye, J. X. Huang, and H. Lin. 2011. Finding a good query-related topic for boosting pseudo-relevance feedback. *J. Am. Soc. Inf. Sci. Technol.* 62, 4 (2011), 748–760.
- [156] Z. Zheng, K. Hui, B. He, X. Han, L. Sun, and A. Yates. 2020. BERT-QE: Contextualized query expansion for document re-ranking. arXiv:2009.07288. Retrieved from <http://arxiv.org/abs/2009.07288>

Received 19 July 2021; revised 21 December 2023; accepted 31 January 2024