Published: 24 March 2022
Accepted: 01 September 2021
Revised: 01 August 2021
Received: 01 March 2021

RESEARCH-ARTICLE

Semantic Models for the First-Stage Retrieval: A Comprehensive Review

JIAFENG GUO, Institute of Computing Technology Chinese Academy of Sciences, Beijing, Beijing, China

YINQIONG CAI, Institute of Computing Technology Chinese Academy of Sciences, Beijing, Beijing, China

YIXING FAN, Institute of Computing Technology Chinese Academy of Sciences, Beijing, Beijing, China

FEI SUN, Alibaba Group Holding Limited, Hangzhou, Zhejiang, China

RUQING ZHANG, Institute of Computing Technology Chinese Academy of Sciences, Beijing, Beijing, China

XUEQI CHENG, Institute of Computing Technology Chinese Academy of Sciences, Beijing, Beijing, China

Open Access Support provided by:

Institute of Computing Technology Chinese Academy of Sciences

Alibaba Group Holding Limited

Semantic Models for the First-Stage Retrieval: A Comprehensive Review

JIAFENG GUO, YINQIONG CAI, and YIXING FAN, Institute of Computing Technology, Chinese Academy of Sciences; University of Chinese Academy of Sciences, China
FEI SUN, Alibaba Group, China
RUQING ZHANG and XUEQI CHENG, Institute of Computing Technology, Chinese Academy of Sciences; University of Chinese Academy of Sciences, China

66

Multi-stage ranking pipelines have been a practical solution in modern search systems, where the first-stage retrieval is to return a subset of candidate documents and latter stages attempt to re-rank those candidates. Unlike re-ranking stages going through quick technique shifts over the past decades, the first-stage retrieval has long been dominated by classical term-based models. Unfortunately, these models suffer from the vocabulary mismatch problem, which may block re-ranking stages from relevant documents at the very beginning. Therefore, it has been a long-term desire to build semantic models for the first-stage retrieval that can achieve high recall efficiently. Recently, we have witnessed an explosive growth of research interests on the first-stage semantic retrieval models. We believe it is the right time to survey current status, learn from existing methods, and gain some insights for future development. In this article, we describe the current landscape of the first-stage retrieval models under a unified framework to clarify the connection between classical term-based retrieval methods, early semantic retrieval methods, and neural semantic retrieval methods. Moreover, we identify some open challenges and envision some future directions, with the hope of inspiring more research on these important yet less investigated topics.

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

Additional Key Words and Phrases: Semantic retrieval models, information retrieval, survey

ACM Reference format:
Jiafeng Guo, Yinqiong Cai, Yixing Fan, Fei Sun, Ruqing Zhang, and Xueqi Cheng. 2022. Semantic Models for the First-Stage Retrieval: A Comprehensive Review. *ACM Trans. Inf. Syst.* 40, 4, Article 66 (March 2022), 42 pages.
https://doi.org/10.1145/3486250

This work was funded by Beijing Academy of Artificial Intelligence (BAAI) under Grants No. BAAI2019ZD0306, the National Natural Science Foundation of China (NSFC) under Grants No. 61902381, 62006218, and 61872338, the Youth Innovation Promotion Association CAS under Grants No. 20144310, and 2021100, the Lenovo-CAS Joint Lab Youth Scientist Project, and the Foundation and Frontier Research Key Program of Chongqing Science and Technology Commission (No. cstc2017jcyjBX0059).
Authors' addresses: J. Guo, Y. Cai, Y. Fan (corresponding authors), R. Zhang, and X. Cheng (corresponding authors), CAS Key Lab of Network Data Science and Technology, Institute of Computing Technology, Chinese Academy of Sciences; University of Chinese Academy of Sciences, NO. 6 Kexueyuan South Road, Haidian District, Beijing, China, 100190; emails: {guojiafeng, caiyinqiong18s, fanyixing, zhangruqing, cxq}@ict.ac.cn; F. Sun, Alibaba Group, Beijing, China, 100102; email: ofey.sf@alibaba-inc.com.

1046-8188/2022/03-ART66 $15.00
https://doi.org/10.1145/3486250

1 INTRODUCTION

Large-scale query-document retrieval is a key problem in search systems, such as Web search engines, which aims to return a set of relevant documents from a large document repository given a user query. To balance the search efficiency and effectiveness, modern search systems typically employ a multi-stage ranking pipeline in practice, as shown in Figure 1. The first-stage retrieval aims to return an initial set of candidate documents from a large repository by some cheaper ranking models assisted by some specially designed indexing structures. Later, several re-ranking stages take more complex and effective ranking models to prune and improve the ranked document list output by the previous stage. Such a "retrieval and re-ranking" pipeline has been widely adopted in both academia [40, 147] and industry [133, 167] and achieved state-of-the-art results on multiple **information retrieval (IR)** benchmarks [59, 160, 205].

Besides the pipeline architecture, to achieve a successful retrieval, it is generally recognized that the system needs to understand the query and the document well so that it can find relevant results to users' information needs. Therefore, semantic models are expected throughout the pipeline but with different requirements and goals at different stages. For the first-stage retrieval, the model aims to recall all potentially relevant documents from the whole collection. Thus, it is desired to build semantic models that can achieve high recall efficiently—that is, to return a subset of documents that contain as many relevant documents as possible within a short time span. For latter re-ranking stages, only a small number of documents are fed into the ranking model. As a result, semantic models used for re-ranking are allowed to employ more sophisticated architectures to achieve high precision—that is, to put as many relevant documents as possible to top positions of the list.

Over the past decades, we have witnessed re-ranking stages going through quick technique shifts toward more and more powerful semantic models, from early probabilistic models [177, 178, 202] to learning to rank models [126, 134] to recent neural ranking models [82, 96, 161]. Specifically, with BERT-style pre-training tasks on cross-attention models, better contextualized representations and deeper interactions between query-document pairs have led to significant improvement on re-ranking effectiveness [161, 163]. However, these models are often quite computationally expensive, which makes them unable to handle high-throughput incoming queries each with a large collection of candidate documents in the first-stage retrieval.

On the contrary, the first-stage retrieval has long been dominated by classical term-based models. Specifically, the discrete symbolic representation (i.e., **bag-of-words (BOW)** representation) is adopted for both queries and documents, and the inverted indexing technique is leveraged to manage large-scale documents. Term-based retrieval models such as BM25 (term matching + TF-IDF weights) are then applied for the first-stage retrieval. Apparently, such term-based models are very efficient due to the simple logic and powerful index. Meanwhile, they have also been demonstrated to achieve reasonable good recall performance in practice [40, 133]. However, there are still clear drawbacks with such term-based models: (1) they may suffer from the *vocabulary mismatch* problem [72, 239] due to the independence assumption, and (2) they may not well capture document semantics by ignoring term ordering information [127]. Due to these limitations, term-based models may play as a "blocker" that prevents re-ranking models from relevant documents at the very beginning. To resolve this problem, continuous efforts have been made over the past decades, including query expansion [119, 124, 171, 217], document expansion [3, 65, 135], term dependency models [76, 148, 218], topic models [54, 213], and translation models for IR [21, 108], among others. However, the research progress on the first-stage retrieval is relatively slow since most of these approaches are still within the discrete symbolic representation paradigm and inherit its limitations inevitably.

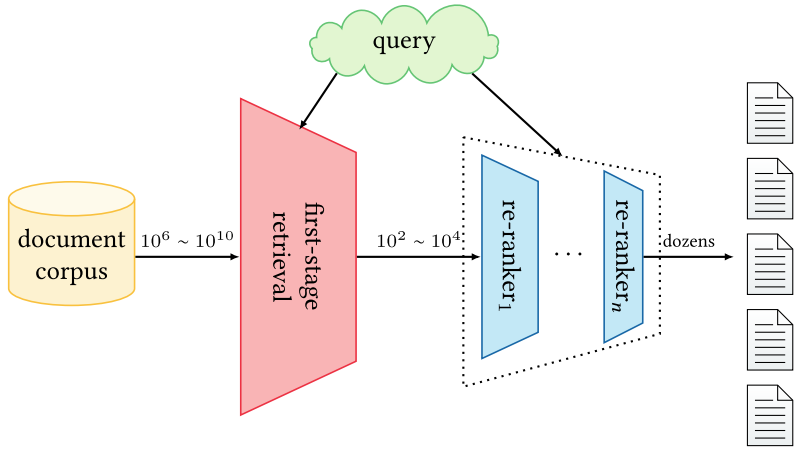


Fig. 1. The multi-stage architecture of modern IR systems.

In recent years, along with the development of representation learning methods in IR, we have witnessed an explosive growth of research interests in the first-stage semantic retrieval models. Since 2013, the rise of the word embedding technique [26, 149, 168] has stimulated a large amount of work on exploiting it for the first-stage retrieval [45, 73, 207]. Unlike the discrete symbolic representation, word embedding is a dense representation that may alleviate the vocabulary mismatch problem to some extent. After 2016, there was a surge of research interest in applying deep learning technique for the first-stage retrieval [28, 87]. These approaches have been studied either to improve document representations within the conventional discrete symbolic representation paradigm [14, 49, 164] or to directly form a new series of semantic retrieval models within the sparse/dense representation paradigm [81, 100, 112, 228]. Since there has been a significant body of works created, we believe it is the right time to survey the current status, learn from existing methods, and gain some insights for future development.

This survey focuses on semantic models for the first-stage retrieval of unstructured texts, referred to as *semantic retrieval models* in the following sections. We describe the current landscape of the first-stage retrieval models under a unified framework to clarify the connection between classical term-based retrieval methods, early semantic retrieval methods, and neural semantic retrieval methods. Specifically, we pay attention to recent neural semantic retrieval methods, summarizing them into three paradigms from the perspective of model architecture, namely sparse retrieval methods, dense retrieval methods, and hybrid retrieval methods. We also refer to key topics about neural semantic retrieval models learning. Moreover, we discuss unresolved challenges and suggest potentially promising directions for future work. The following should be noted. First, some studies also refer to the first-stage retrieval a *ranking stage*, a *search stage*, or a *recall stage*. In this survey, we will refer to it as the *retrieval stage* for consistency and simplicity. Second, the survey mainly focuses on ranking algorithms of semantic retrieval models, and thus it will only briefly mention indexing methods. Readers who are interested in sparse or dense indexing techniques could refer to other works [38, 157, 235, 241].

As far as we know, this is the first survey on both traditional and neural semantic models for the first-stage retrieval. It reviews early semantic retrieval models proposed from 1990 to 2013 and covers neural semantic retrieval models published in major conferences (e.g., ACL, ICLR, AAAI, SIGIR, TheWebConf, CIKM, WSDM, EMNLP, and ECIR) and journals (e.g., TOIS, TKDE, TACL, and IP&M) in the fields of deep learning, **natural language processing (NLP),** and IR from 2013 to

June 2021. There have been some surveys on neural models for IR [83, 151, 152, 165], but none of them focused on the first-stage retrieval. For example, Onal et al. [165] paid attention to the application of neural methods to different IR tasks. Guo et al. [83] took a deep look into deep neural networks for re-ranking stages. For the first-stage retrieval, the booklet by Li and Xu [127] talked about early semantic retrieval models, but without recent booming neural models for the first-stage retrieval. Recently, Lin et al. [131] discussed several pre-training models for the first-stage retrieval and re-ranking stages. Different from them, we make an comprehensive overview of semantic models for the first-stage retrieval under a unified framework, including early semantic retrieval models, neural semantic retrieval models, and the connection between them.

To sum up, our contributions include the following:

(1) We describe the current landscape of the first-stage retrieval models under a unified framework to clarify the connection between the classical term-based retrieval, early methods for semantic retrieval, and neural methods for semantic retrieval.
(2) We provide a comprehensive and up-to-date review of semantic retrieval models, with a brief review of early semantic retrieval models and a detailed description of recent neural semantic retrieval models.
(3) We summarize neural semantic retrieval models into three paradigms from the perspective of model architecture: sparse retrieval methods, dense retrieval methods and hybrid retrieval methods. We also discuss key topics on model learning, including loss functions and negative sampling strategies.
(4) We discuss some open challenges and suggest potentially promising directions for future works.

We organize this survey as follows. We first introduce three typical applications of semantic retrieval models in Section 2. Then, we provide some background knowledge, including problem formalization, index methods, and classical term-based retrieval methods in Section 3. We sketch early methods for semantic retrieval in Section 4. In Section 5, we review existing neural methods for semantic retrieval from the perspective of model architecture and introduce key topics on model learning. Finally, we discuss challenges and future directions in Section 6 and conclude this survey in Section 7.

2 MAJOR APPLICATIONS OF SEMANTIC RETRIEVAL MODELS

The first-stage retrieval plays an essential role in almost all large-scale IR applications. In this section, we describe three major text retrieval applications: ad hoc retrieval [12], **open-domain question answering (OpenQA)** [191, 206], and **community-based question answering (CQA)** [31, 193].

Ad hoc retrieval is a typical retrieval task, and there has been a long research history on ad hoc retrieval models. In this task, users express their information needs as queries, then trigger searches in the retrieval system to obtain relevant documents. All retrieved documents are often returned as a ranked list according to the degree of relevance to the user query. A major characteristic of ad hoc retrieval is the length heterogeneity between the query and the document. Queries are often short in length, consisting of only a few keywords [151], whereas documents have longer texts, ranging from multiple sentences to several paragraphs. Such heterogeneity between queries and documents leads to the classical vocabulary mismatch problem, which has been a long-term challenge in both the retrieval stage as well as re-ranking stages in ad hoc retrieval [127]. The earliest datasets to support reliable evaluation of the first-stage retrieval models are always based on TREC collections, such as Associated Press Newswire (AP), Wall Street Journal (WSJ), and Robust [117]. The number of documents in these collections is usually

in the hundreds of thousands, and documents are usually news articles. Later, larger collections based on Web data, such as ClueWeb [44], were built for the evaluation of retrieval technology. However, the number of queries in these datasets is only a few hundred, which is not enough for the training of neural-based retrieval models. In recent years, large-scale datasets, such as MS MARCO [160], TREC CAR [59], and TREC Deep Learning Track [47], have been released, which label relevant documents for hundreds of thousands of queries. The availability of these large-scale datasets has greatly promoted the development of neural retrieval models. In addition, there are some domain-specific retrieval datasets, such as GOV2 [43], TREC Medical Records Track (MedTrack), and TREC-COVID [203], which are also commonly used for the evaluation.

OpenQA is a task to answer any sort of (factoid) questions that humans might ask, using a large collection of documents (e.g., Wikipedia or Web page) as the information source [110]. Unlike the ad hoc retrieval that aims to return a ranked list of documents, the OpenQA task is to extract a text span as the answer to the question. To achieve this, most existing works build the OpenQA system as a two-stage pipeline [36]: (1) a *document retriever* selects a small set of relevant documents that probably contain the answer from a large-scale collection; (2) a *document reader* extracts the answer from relevant documents returned by the document retriever. In our work, we only consider the document retriever component since the document reader is outside of the scope of this work. Typically, the question in OpenQA tasks is a natural language sentence, which has well-formatted linguistic structures, whereas the document is often a small snippet of text, ranging from several sentences to a passage [56, 63]. Moreover, relevant documents are required to be not only topically related to but also correctly address the question, which requires more semantics understanding except for exact term matching features. For the evaluation of the first-stage retrieval models on OpenQA tasks, several benchmark datasets are available. The most commonly used datasets, such as SQuAD-open [36], SearchQA [63], TriviaQA-unfiltered [107], and Natural Questions Open [116], have tens of thousands of queries for model training. Several smaller-scale datasets, such as WebQuestions [20] and CuratedTREC [16], are also often used for model evaluation. The document collection in these datasets is usually based on Wikipedia pages (e.g., SQuAD-open and Natural Questions Open) or Web pages (e.g., SearchQA and WebQuestions), and queries are written by crowd-workers (e.g., SQuAD-open) or crawled from existing websites (e.g., SearchQA and TriviaQA-unfiltered).

CQA aims to address the user's questions using the archived **question-answer (QA)** pairs in the repository, since CQA systems have already accumulated a large amount of high-quality human-generated QA pairs, such as Yahoo! Answers,[1] Stack Overflow,[2] and Quora.[3] There are two different ways to produce the answer to a user's question. One is to directly retrieve answers from the collection if the answer exists [208]. The other is to select the duplicate question from the collection and take the accompanied answer as the result [212]. Both of these two ways require the retrieval system to first recall a subset of candidates from the whole collection and then re-rank candidates to generate the final result. However, targets (i.e., answers and questions) in these two ways often have very different expressions, leading to different challenges in terms of semantic modeling. First, the duplicate question retrieval needs to capture semantic similarities between words (phrases) since there are often different ways to express the same question. Second, the answer retrieval needs to model logical relationships between questions and answers. Although many datasets are constructed based on CQA data, few of them are suitable for evaluating the

[1] https://answers.yahoo.com.
[2] http://www.stackoverflow.com/.
[3] http://www.quora.com/.

first-stage retrieval models. Existing related works usually conduct experiments on QQP[4] and WikiAnswers [66] datasets.

There are also some other retrieval scenarios, such as entity linking [80], e-commerce search [125, 128, 234], and sponsored search [68]. For these applications, academic researchers and industrial developers have realized the importance of utilizing semantic information for the first-stage retrieval. Due to page limitations, we will not discuss these works in this survey, but it is possible and necessary to generalize techniques applied in text retrieval to other retrieval tasks.

3 BACKGROUND

In this section, we first characterize the first-stage retrieval by giving a unified formulation of the first-stage retrieval models. Then, we introduce typical indexing methods cooperating retrieval models to support efficient retrieval. Finally, we summarize classical term-based retrieval methods.

3.1 Problem Formalization

Given a query q, the first-stage retrieval aims to recall all potentially relevant documents from a large corpus $C = \{d_1, d_2, \ldots, d_N\}$. Different from re-ranking stages with a small set of candidates, the corpus size N for the first-stage retrieval can range from millions (e.g., Wikipedia) to billions (e.g., the Web). Thus, efficiency is a crucial concern for models used in the first-stage retrieval.

Formally, given a dataset $\mathcal{D} = \{(q_i, D_i, Y_i)\}_{i=1}^{n}$, where q_i denotes a user query, $D_i = [d_{i1}, d_{i2}, \ldots, d_{ik}]$ denotes a list of documents to the query q_i, and $Y_i = [y_{i1}, y_{i2}, \ldots, y_{ik}] \in \{1, 2, \ldots, l\}$ is the corresponding relevance label of each document in D_i. There exists a total order between relevance labels $l > l-1 > \cdots > 1$, where $>$ denotes the order relation. Note here that the number of labeled documents k to each query is often significantly smaller than the corpus size N, since it is impossible to manually annotate all the huge amount of documents. The goal of the first-stage retrieval is to learn a model $s(\cdot, \cdot)$ from $\mathcal{D}$ that gives high scores to relevant (q, d) pairs and low scores to irrelevant ones. For any query-document pair (q, d), $s(q, d)$ gives a score that reflects the relevance degree between q and d, and thus allows one to rank all the documents in the corpus C according to predicted scores. Without loss of generality, the scoring function can be abstracted by the following unified formulation:

$$s(q, d) = f\Big(\phi(q), \psi(d)\Big), \tag{1}$$

where $q \in X$ and $d \in Y$ are the input query and document, and two representation functions $\phi : X \to \mathbb{R}^{k_1}$ and $\psi : Y \to \mathbb{R}^{k_2}$ map a sequence of tokens in X and Y to their associated embeddings $\phi(q)$ and $\psi(d)$, respectively. To build a responsive model for the first-stage retrieval, it leads to a number of requirements on these three components:

- The document representation function ψ should be independent of the query since queries are unknown before the search system is deployed. In this way, document representations can be pre-computed and indexed offline with methods in Section 3.2. Meanwhile, this means that the $\psi(d)$ component can be sophisticated to some extent since it has no impact on the online serving.
- The query representation function ϕ is required to be as efficient as possible since it needs to compute query embeddings online. Thanks to the nature of independence, two components ϕ and ψ can be identical or different, which is flexible enough to design models for different retrieval tasks with homogeneous or heterogeneous inputs.

[4] https://quoradata.quora.com/First-Quora-Dataset-Release-Question-Pairs.

- To satisfy the real-time retrieval requirement, on the one hand, the scoring function f should be as simple as possible to minimize the amount of online computation, and on the other hand, it must take the indexing method into account.

3.2 Indexing Methods

As mentioned previously, one major difference between the first-stage retrieval and re-ranking stages is that the former does ranking on large-scale documents in the repository. Thus, the efficiency of the first-stage retrieval models is one of the core considerations. In practice, to support storing and fast retrieval of documents in the whole repository, retrieval systems need to build an index, where the indexing technique is crucial to the rapid response during the online serving. There are many indexing techniques, such as signature, inverted index, and dense vector index. Rather than exploring all the existing approaches, we only describe the fundamental principle of two typical indexing schemes.

The inverted index is currently the most popular indexing scheme and is used for many applications due to its simplicity and efficiency. Before building an inverted index, each document in the collection is parsed and segmented into a list of tokens. Then, the inverted index is created, which mainly consists of a dictionary and a collection of posting lists. The dictionary includes all the terms found in the collection and their document frequencies. Each posting list records document identifiers, term occurrence frequencies, and possibly other information of documents in which the corresponding term appears. During the online serving, for a user's query, the top k most similar documents are fetched in turn with the help of the inverted index. Concretely, the query is processed with one term at a time. Initially, each document has a similarity of zero to the query. Then, for each query term t, the similarity score of each document in t's posting list increases by the contribution of t to the similarity of the query-document pair. Once all query terms have been processed, the k largest similarity scores are identified, and the corresponding document list is returned to the user. In fact, many acceleration strategies are applied during the process to improve retrieval efficiency, but they are omitted here. More details about the inverted index technique could be found in other works [214, 241].

Along with the development of neural representation learning methods, a dense vector index based on **approximate nearest neighbor (ANN)** search algorithms is used to support the new representation paradigm. One of reasons the inverted index works well is that the documents' term matrix is very sparse. However, most semantic retrieval models produce dense and distributed document representations, and thus the inverted index method is no longer feasible to retrieve documents efficiently from a large collection. From Equation (1), the retrieval problem could be viewed as the nearest neighbor search problem [188], once the query embedding and all the document embeddings have been calculated. This fundamental problem has been well studied in the research community [1, 8]. The simplest approach to the nearest neighbor search is the brute-force search, which scans all the candidates and computes similarity scores one by one. However, the brute-force search becomes impractical when the size of collections exceeds a certain point. Thus, most research resorts to an ANN search [11, 64, 129], which allows for a slight loss in precision while yielding multiple orders of magnitude improvement in speed. Generally, existing ANN search algorithms can be categorized into four major types: tree-based [17, 19], hashing-based [53, 99], quantization-based [79, 102], and proximity graph approaches [113, 144]. The earliest solutions to ANN search are based on locality-sensitive hashing [99], but currently proximity graph methods [113, 144] yield a better performance among all the approaches in most respects based on a popular benchmark.[5] Graph-based methods build the index by retaining the neighborhood

[5] http://ann-benchmarks.com/.

information for each individual data point toward other data points or a set of pivot points. Then, various greedy heuristics are proposed to navigate the proximity graph for a given query point. So far, several open source libraries for ANN search, such as Faiss [105] and SPTAG [39], have been developed, and search engines[6,7,8] supporting ANN search have been built and applied widely.

3.3 Classical Term-Based Retrieval

This section provides an overview of classical term-based methods for the first-stage retrieval, including the **vector space model (VSM)**, probabilistic retrieval models, and language models for IR. In general, these methods build representations of queries and documents based on the BOW assumption where each text is represented as a bag (multiset) of its words, disregarding grammar and even word order. Particularly, the representation functions ϕ and ψ are set to be manually defined feature functions, such as word frequency, and dimensions of representations (i.e., k_1 and k_2) are generally equal to the vocabulary size. The representation functions ϕ and ψ are usually different for queries and documents, but they all pledge the sparsity of representations so that the inverted index could be used to support efficient retrieval.

The early representative of term-based methods is the VSM [185], which represents queries and documents as high-dimensional sparse vectors in a common vector space. Under this framework, queries and documents are viewed as vectors with each dimension corresponding to a term in the vocabulary, where the weight of each dimension can be determined by different functions,such as **term frequency (TF)**, **inverse document frequency (IDF)**, or the composite of them [183, 184]. Then, one can use the similarity (usually cosine similarity) between a query vector and a document vector as the relevance measure of the query-document pair. The resulting scores can then be used to select the top relevant documents for the query. The VSM has become the fundamental of a series of IR solutions—the probabilistic retrieval model and language model for IR can be both viewed as the instantiation of VSM with different weighting schemes.

Probabilistic methods are one of the oldest formal models in IR, which introduce the probability theory as a principled foundation to estimate the relevance probability. **The binary independence model (BIM)** [178] is the most original and influential probabilistic retrieval model. It represents documents and queries to binary term vectors, that an entry is 1 if the corresponding term occurs in the document, and otherwise the entry is 0. With these representations, "binary" and "term independency" assumptions are introduced by the BIM. But these assumptions are contrary to facts, so a number of extensions are proposed to relax some assumptions of the BIM, such as the tree dependence model [202] and BM25 [177]. In particular, BM25 takes into account the document frequency, document length and TF, which has been widely used and quite successful across different academic research as well as commercial systems [133, 167].

Instead of modeling the relevance probability explicitly, language models (LM) for IR [170] build a language model M_d for each document d, then documents are ranked based on the likelihood of generating the query q (i.e., $P(q|M_d)$). The document language model is also built on the BOW assumption and could be instantiated as either multiple Bernoulli [170] or multinomial [89, 150]. Experimental results in the work of Ponte and Croft [170] prove the effectiveness of term weights coming from language models over traditional TF-IDF weight. Moreover, language models provide another perspective for modeling retrieval tasks and inspire many extended approaches [29, 229].

In summary, modeling relevance in a shallow lexical way, especially combined with the inverted index, endows classical term-based models a key advantage on efficiency, making it possible to

[6] https://www.elastic.co/cn/elasticsearch/.
[7] https://vespa.ai/.
[8] https://milvus.io/.

retrieve from billions of documents quickly. However, such a paradigm is also accompanied by clear drawbacks, like the well-known vocabulary mismatch problem or not well capturing text semantics. Therefore, more sophisticated semantic models for improving the first-stage retrieval performance have started to attract researchers' interests and are discussed in the following.

4 EARLY METHODS FOR SEMANTIC RETRIEVAL

From the 1990s to the 2000s, extensive studies have been carried out to improve term-based retrieval methods. Most of them mine information from external resources or the collection itself to enrich query representations $\phi(q)$, document representations $\psi(d)$, or both of them for semantic retrieval. Here, we sketch a brief picture of some of them.

4.1 Query Expansion

To compensate for the mismatch between queries and documents, the query expansion technique is used to expand the original query with terms selected from external resources [217]. In this way, query representations $\phi(q)$ are enriched, and more documents could be considered during the retrieval process through the extended query terms.

Query expansion is the process of adding relevant terms to a query to improve retrieval effectiveness. There are a number of query expansion methods, and they can be classified into global methods [124, 171] and local methods [2, 230]. Global methods expand or reformulate query words by analyzing word co-occurrences from the corpus being searched or using an external handcrafted thesaurus (e.g., WordNet) [204]. Although a number of data-driven query expansion methods (e.g., [13]) can improve the average retrieval performance, they are shown to be unstable across queries. However, local methods adjust a query based on top-ranked documents retrieved by the original query. This kind of query expansion is called **pseudo-relevance feedback (PRF)** [33], which has been proven to be highly effective to improve the performance of many retrieval models [138, 179]. The relevance model [119], the mixture model, and the divergence minimization model [230] are the first PRF methods proposed under the language modeling framework. Since then, several other local methods have been proposed, but the relevance model remains among state-of-the-art PRF methods and performs more robustly than many other methods [138].

In general, query expansion methods have been widely studied and adopted in IR applications, but they do not always yield a consistent improvement. Especially expansion methods based on PRF are prone to the query drift problem [46]. Subsequently, with the development of the deep learning technique, neural word embeddings and deep language models are used to enhance query expansion methods [58, 146, 181].

4.2 Document Expansion

An alternative to query expansion is to perform the expansion for all documents in the corpus, then those enriched documents are indexed and searched as before. Intuitively, document expansion methods supplement each posting list in the inverted index, which have shown to be particularly effective for IR tasks [3, 65, 200].

Document expansion is first proposed in the speech retrieval community [192]. Singhal and Pereira [192] proposed to use the original document as a query into the collection, and the 10 most relevant documents were selected. Then, they enhanced the representation of the original document by adding to the document vector a linearly weighted mixture of related documents. Similarly, Efron et al. [65] followed a similar approach on short text retrieval tasks. They submitted documents as pseudo-queries and performed document expansion based on the analysis of the result set. Different from the retrieval-based method to determine related documents for expansion, it is another way to use document clustering to determine similar documents, and document

expansion is carried out with respect to these results [114, 135]. Both works report significant improvements over non-expanded baselines on the TREC ad hoc document retrieval task. In addition to using the document collection itself, it is also helpful to use external information to augment document representations [3, 190]. For example, Agirre et al. [3] presented a novel document expansion method based on a WordNet-based system to find related concepts and words, which is the first to perform document expansion using lexical semantic resources.

The document expansion technique has been less popular with IR research because it is less amenable to rapid experiments. The corpus needs to be re-indexed every time the expansion technique changes, which is a costly process. In contrast, manipulations to query representations can happen at retrieval time and hence are much faster. In addition, the success of document expansion has been mixed. Billerbeck and Zobel [23] explored both query expansion and document expansion in the same framework and concluded that the former is consistently more effective. Nevertheless, dramatic improvement for the first-stage retrieval has been achieved after equipping the document expansion technique with neural models, such as doc2query [164] and docTTTTTquery [162] (see Section 5.1).

4.3 Term Dependency Models

Typically, term-based methods consider terms in the document independently and ignore the term orders. As a result, concepts represented by multiple contiguous words cannot be depicted correctly, and the stronger relevance of consecutive or ordered terms matching between queries and documents cannot be reflected well. Term dependency models attempt to address the preceding problem by incorporating term dependencies into the representation functions ϕ and ψ.

A natural way is to extend the dictionary in the inverted index with frequent phrases. For example, Fagan [67] tried to incorporate phrases into the VSM, where phrases are viewed as additional dimensions in the representation space. Then, the scoring function can be formalized to the combination of term-level score and phrase-level score:

$$s(q,d) = w_{\text{term}} \cdot \underbrace{s_{\text{term}}(q,d)}_{\text{term score}} + w_{\text{phr}} \cdot \underbrace{s_{\text{phr}}(q,d)}_{\text{phrase score}}, \tag{2}$$

where w_{term} and w_{phr} are weights to achieve weight normalization, and the score of a phrase can be defined as the average of TF-IDF weights of its component terms. Xu et al. [218] also investigated the approach that extends BM25 with n-grams. They defined the BM25 kernel as follows:

$$\text{BM25-Kernel}(q,d) = \sum_t \text{BM25-Kernel}_t(q,d), \tag{3}$$

where $\text{BM25-Kernel}_t(q,d)$ denotes the BM25 kernel of type t, and t can be bigram, trigram, and so on.

$$\text{BM25-Kernel}_t(q,d) = \sum_x \text{IDF}_t(x) \times \frac{(k_3+1) \times f_t(x,q)}{k_3 + f_t(x,q)} \times \frac{(k_1+1) \times f_t(x,d)}{k_1 \left(1 - b + b\frac{f_t(d)}{\bar{f}_t}\right) + f_t(x,d)}, \tag{4}$$

where x denotes a n-gram of type t, $f_t(x,q)$ and $f_t(x,d)$ are frequency of unit x in query q and document d, respectively, $f_t(d)$ is total number of units with type t in document d, $\bar{f}_t$ is the average number of $f_t(d)$ within the whole collection, and $k1$, $k3$, and b are parameters.

Integrating term dependencies to term-based methods increases the complexity, but gains are not significant as expected [118]. The Markov random field approach proposed by Metzler and Croft [148] reports the first clear improvement for term dependency models over term-based baselines. In the Markov random field approach, the document and each term in the query are represented as a node, respectively. The document node is connected to every query term node.

Moreover, there are some edges between query term nodes, based on pre-defined dependency relations (e.g., bigram, named entity, or co-occurrence within a distance), to represent their dependencies. Then, the joint probability of query q and document d can be formally represented as

$$P(q,d) = \frac{1}{Z} \prod_{c \in \text{clique(G)}} \exp\Big(\lambda_c f(c)\Big), \tag{5}$$

where c denotes a clique on the constructed graph G, λ_c is the interpolation coefficient, $f(c)$ is the potential function defined on clique c, and Z denotes the partition function. In practice, we can define different feature functions to capture different types of term dependencies, and the coefficient λ_c can be optimized toward designative retrieval metrics, as in the work of Metzler and Croft [148].

Although those methods are capable of capturing certain syntactics and semantics, their "understanding" capability is much limited. How to go beyond these simple counting statistics and mine deeper signals to better query-document matching is still an open question. Nevertheless, there is no doubt that term dependency models demonstrate the importance of understanding document semantics with context, stimulating a series of neural retrieval models that emphasize the capturing of contextual information [112, 232].

4.4 Topic Models

Another line to improve ϕ and ψ simultaneously focuses on semantic relationships between words—usually modeling words' co-occurrence relation to discover latent topics in texts and matching queries and documents by their topic representations. In this way, each dimension of the representation indicates a topic instead of a term. In addition, the inverted index becomes impractical since topic representations lose sparsity.

Topic modeling methods have received much attention in NLP tasks. Overall, they can be divided into two categories: probabilistic and non-probabilistic approaches. The non-probabilistic topic model, such as **latent semantic indexing (LSI)** [54], non-negative matrix factorization [121], and regularized LSI [211], is usually obtained by matrix factorization. Taking LSI as an example, it uses a truncated singular value decomposition (SVD) to obtain a low-rank approximation to the document-term matrix, then each document can be represented as a mixture of topics. Other topic models choose different strategies to conduct the matrix factorization. For example, non-negative matrix factorization introduces the non-negative constraint and regularized LSI assumes topics are sparse. For probabilistic approaches, probabilistic LSI [90] and latent Dirichlet allocation [25] are most widely used. Probabilistic topic models are usually generative models, where each topic is defined as a probabilistic distribution over terms in the vocabulary and each document in the collection is defined as a probabilistic distribution over topics.

Studies that apply topic models to improve retrieval results can be classified in two ways. The first is to obtain query and document representations in the topic space, then calculate relevance scores based on topic representations. For example, the LSI learns a linear projection that casts the sparse BOW text vector into a dense vector in latent topic space, then the relevance score between a query and a document is the cosine similarity of their corresponding dense vectors. In the latent Dirichlet allocation based retrieval model [213], queries and documents are represented by their latent topic distributions. The relevance score of each query-document pair is computed by the Kullback-Leibler divergence as follows:

$$s(q,d) = 1 - \frac{1}{2}\Big(\text{KL}\left(v_q \| v_d\right) + \text{KL}\left(v_d \| v_q\right)\Big) = 1 - \frac{1}{2}\sum_{k=1}^{K}\left(\left(v_q^k - v_d^k\right)\log\frac{v_q^k}{v_d^k}\right), \tag{6}$$

where v_q and v_d are topic representations of query q and document d, respectively, and v_q^k and v_d^k are the k-th element of v_q and v_d.

Another way is to combine topic models with term-based methods. A simple and direct approach is to linearly combine relevance scores calculated by topic models and term-based models [90]:

$$s(q,d) = \alpha s_{\text{topic}}(q,d) + (1-\alpha) s_{\text{term}}(q,d), \tag{7}$$

where α is the coefficient, and $s_{\text{topic}}(q,d)$ and $s_{\text{term}}(q,d)$ are the topic matching score and term matching score, respectively. In addition, probabilistic topic models can be taken as the smoothing method to language models for IR [57, 213, 224].

$$P(q|d) = \prod_{w \in q} P(w|d) = \prod_{w \in q} \Big(\alpha P_{\text{LM}}(w|d) + (1-\alpha) P_{\text{TM}}(w|d) \Big), \tag{8}$$

where α is the coefficient, $P_{\text{LM}}(w|d)$ and $P_{\text{TM}}(w|d)$ are generating probabilities of word w given document d estimated by a language model and a topic model. The $P_{\text{TM}}(w|d)$ can be defined as

$$P_{\text{TM}}(w|d) = \sum_{z=1}^{K} P(w|z) P(z|d), \tag{9}$$

where z denotes a latent topic.

According to results in the work of Atreya and Elkan [10], using latent topic representations obtained by topic models alone for IR tasks only has small gains or poor performance over term-based baselines, unless combining them with term-based methods. Possible reasons include the following. First, these topic models are mostly unsupervised, learning with a reconstruction objective, either based on mean squared error [54] or likelihood [25, 90]. They may not learn a matching score that works well for specific retrieval tasks. Second, word co-occurrence patterns learned by these topic models are from documents, ignoring the fact that language usage in searching texts (queries) can be different from those in writing texts (documents), especially when the heterogeneity between queries and documents is significant. Third, topic models represent documents as compact vectors, losing detailed matching signals over term-level. Later, using more powerful neural models (e.g., doc2vec [120]), instead of topic models for IR has achieved better results [5, 6].

4.5 Translation Models

A notable attempt to address the vocabulary mismatch problem is the statistical translation approach, which enriches the document representation function ψ from TF to translation models. **Statistical machine translation (SMT)** is leveraged for IR by viewing queries as texts in one language and documents as texts in another language. Retrieval by translation models needs to learn translation probabilities from queries to associated relevant documents, which can be obtained from labeled data, and thus belongs to the supervised learning approach.

Berger and Lafferty [21] first proposed to formulate retrieval tasks as an SMT problem, in which query q is translated into document d with the conditional probability $P(d|q)$. The model can be written as

$$P(d|q) \propto P(q|d) P(d), \tag{10}$$

where $P(q|d)$ denotes a translation model that translates d to q, and $P(d)$ denotes a language model giving rise to d. Translation probabilities can be estimated with queries and their associated relevant documents (e.g., click-through datasets), and the language model can be learned with different schemes, such as BM25. As Karimzadehgan and Zhai [108] have noted, the translation probability $P(q|d)$ allows for the incorporation of semantic relations between terms with non-zero probabilities, which provides a sort of "semantic smoothing" for $P(q|d)$.

One important difference between conventional machine translation and machine translation for retrieval is that both queries (target language) and documents (source language) are in the same language. The probability of translating a word to itself should be quite high—that is, $P(w \mid w) > 0$, which corresponds to exact term matching in retrieval tasks. How to accurately calculate self-translation probabilities is an important issue. If self-translation probabilities are too large, it will make other translation probabilities small and decrease the effect of using translation. However, if self-translation probabilities are too small, then it will make exact matching less effective and hurt the performance of retrieval. A number of methods [74, 108, 109] have been proposed to estimate self-translation probabilities. For example, Karimzadehgan and Zhai [108] proposed to address this estimation problem based on normalized mutual information between words, which is less computationally expensive and has better coverage of query words than the synthetic query method of estimation [21]:

$$P_{mi-\alpha} = \begin{cases} \alpha + (1-\alpha)P_{mi}(w \mid u) & \text{if } w = u \\ (1-\alpha)P_{mi}(w \mid u) & \text{if } w \neq u \end{cases}, \tag{11}$$

where α is the weight that is empirically set on held-out data. Similarly, an alternative heuristic is to impose constant self-translation probabilities for all words in the vocabulary [109]—that is, setting $P(u|u)$ to a constant value s for every u, where $P_t(w|u)$ is estimated according to

$$P_{mi-s} = \begin{cases} s & \text{if } w = u \\ (1-s)\frac{P_{mi}(w|u)}{\sum_{v \neq u} P_{mi}(v|u)} & \text{if } w \neq u \end{cases}. \tag{12}$$

All these methods assume that self-translation probabilities estimated directly from data are not optimal for retrieval tasks, and the authors have demonstrated that significant improvement can be achieved by adjusting the probabilities [74].

Statistical translation models have also been applied to query expansion. For example, Riezler and Liu [176] suggested utilizing a word-based translation model for query expansion. The model is trained with click-through data consisting of queries and snippets of clicked Web pages. Gao and Nie [75] generalized the word-based translation model to a concept-based model and employed the model in query expansion.

Nevertheless, SMT models have not been used much because they are difficult to train due to data sparsity, and they are not more effective than the term-based retrieval with PRF [119] in most situations. Subsequently, after the appearance of neural word embeddings, using distributed representations to calculate translation probabilities and improve translation models are proposed naturally [73, 242].

Takeaway. Early semantic retrieval models, such as query expansion, document expansion, term dependency models, topic models, and translation models, aim to improve classical BOW representations with semantic units extracted from external resources or the collection itself. Most of them still follow classical term-based methods by representing texts with high-dimensional sparse vectors in symbolic space, so as to be easily integrated with the inverted index to support efficient retrieval. However, these approaches always rely on handcrafted features to build representation functions. As a result, only shallow syntactic and semantic information can be captured. Nevertheless, these early proposals are crucial because they have initially explored beneficial factors for the first-stage retrieval. Thereby, a series of new semantic retrieval models could be inspired when the deep learning technique breaks out, and exciting results could be obtained concomitantly.

5 NEURAL METHODS FOR SEMANTIC RETRIEVAL

Over the past decade, big data and fast computer processors have brought a new era for deep learning technique. A set of simple math units, called *neurons*, are organized into layers and stacked into neural networks. Neural networks have the expressive power to represent complex functions and fit hidden correlations in complicated tasks [98]. For example, it converts discrete symbols (e.g., words, phrases, and sentences) into low-dimensional dense vectors that are able to capture semantic and syntactic features for various NLP tasks [48, 223]. Naturally, it also attracts researchers from the IR field and leads to the research wave of neural approaches to IR (neural IR). However, most earlier researches focus on re-ranking stages [82, 96]. Until recently, much attention was paid to explore neural networks to improve the semantic matching for the first-stage retrieval.

Different from early semantic retrieval models, neural semantic retrieval models employ neural networks to build the representation functions (i.e., ϕ and/or ψ) as well as the scoring function (i.e., f). In this way, these models can learn deep semantics and complex interactions from data in an end-to-end way. From the perspective of model architecture, neural methods for semantic retrieval can be categorized into three classes: *sparse retrieval methods*, *dense retrieval methods*, and *hybrid retrieval methods*. In this section, we will review major works about them. Table 1 summarizes surveyed neural semantic retrieval models in different categories.

5.1 Sparse Retrieval Methods

Sparse retrieval methods usually represent each document and each query with sparse vectors, where only a small number of dimensions are active. The sparse representation has attracted great attention, as it connects to the nature of human memories and shows better interpretability [14]. In addition, sparse representations can be easily integrated into existing inverted indexing engines for efficient retrieval. Without loss of generality, sparse retrieval methods can be categorized into two classes. One is to encode queries and documents still in the symbolic space but employ neural models to improve term weighting schemes, namely *neural weighting schemes*. The other is to directly learn sparse representations (i.e., $\phi(q)$ and $\psi(d)$) in latent space for queries and documents with neural networks, which we call *sparse representation learning*.

5.1.1 Neural Weighting Schemes. One of basic methods to leverage the advantage of neural models while still employing sparse term-based retrieval is to re-weight the term importance before indexing. To this purpose, a direct way is to design neural models to predict term weights based on semantics rather than pre-defined heuristic functions. An alternative method is to augment each document with additional terms, then expanded documents are stored and indexed with classical term-based methods.

One of the earliest methods to learn term weights is the DeepTR model [240], which leverages neural word embeddings to estimate the term importance. Specifically, it constructs a feature vector for each query term and learns a regression model to map feature vectors onto ground truth weights of terms. Estimated weights can be directly used to replace classical term weighting schemes in the inverted index (e.g., BM25 and LM) to generate BOW query representations to improve the retrieval performance. More recently, Frej et al. [71] proposed a term discrimination values learning method, which replaces the IDF field in the original inverted index based on FastText [26]. In addition to the pairwise ranking objective, they also minimized the ℓ_1-norm of BOW document representations to reduce the memory footprint of the inverted index and speed up the retrieval process. In addition, Zuccon et al. [242] used word embeddings within the translation language model for IR. They leveraged word embeddings to estimate translation probabilities between words. This language model captures implicit semantic relations between words in queries

Table 1. Overview of Neural Methods for Semantic Retrieval

	Model		Task		
	Type	Representative Work	Ad Hoc Retrieval	OpenQA	CQA
Sparse Retrieval Methods	Neural Weighting Schemes	DeepTR [240]	√		
		NTLM [242]	√		
		TVD [71]	√		
		DeepCT [49, 51]	√		
		HDCT [50]	√		
		Mitra et al. [156]	√		
		Mitra et al. [154]	√		
		GAR [146]		√	
		doc2query [164]	√		
		docTTTTTquery [162]	√		
		UED [219]	√		
		SparTerm [14]	√		
		DeepImpact [145]	√		
	Sparse Representation Learning	Semantic Hashing [182]	√		
		SNRM [228]	√		
		UHD-BERT [100]	√		
		Ji et al. [103]	√		
Dense Retrieval Methods	Term-Level Representation Learning	OoB [111]	√		
		DESM [155]	√		
		DC-BERT [237]		√	
		ColBERT [112]	√		
		COIL [77]	√		
		De-Former [34]		√	
		PreTTR [141]	√		
		PIQA [186]		√	
		DenSPI [187]		√	
		SPARC [122]		√	
		MUPPET [69]		√	
	Document-Level Representation Learning	FV [45]	√		
		Gillick et al. [81]			√
		Ai et al. [5]	√		
		NVSM[85]	√		
		SAFIR [4]	√		
		Liu et al. [136]	√		
		Tamine et al. [197]	√		
		Henderson et al. [87]		√	
		DPR [110]		√	
		RepBERT [232]	√		
		Lin et al. [132]	√		
		Tahami et al. [196]			
		DSSM [96]	√		
		ARC-I [93]	√		
		QA_LSTM [198]	√		
		ORQA [123]		√	
		REALM [84]		√	
		Chang et al. [35]		√	
		Liang et al. [130]	√	√	
		Poly-encoders [97]	√	√	
		ME-BERT [137]	√	√	
		Tang et al. [199]	√	√	
Hybrid Retrieval Methods		Vulić and Moens [207]	√		
		GLM [73]	√		
		$DESM_{MIXTURE}$ [155]	√		
		Roy et al. [180]	√		
		BOW-CNN [62]			√
		EPIC [142]	√		
		DenSPI [187]		√	
		SPARC [122]		√	
		Hybrid [137]	√	√	
		CLEAR [78]	√		
		Kuzi et al. [115]	√		

and those in relevant documents, thus bridging the vocabulary mismatch and producing more accurate estimations of document relevance.

In recent years, contextual word embeddings, which are often learned with pre-trained language models, have achieved great success in many NLP tasks [55, 169, 221]. Compared with static word embeddings (e.g., Word2Vec [149], GloVe [168], and FastText [26]), contextual word embeddings model the semantic information of words under the global context. There are also several works trying to utilize contextual word embeddings to estimate term weights. For example, Dai and Callan [49, 51] proposed a BERT-based framework (DeepCT) to evaluate the term importance of sentences/passages in a context-aware manner. It maps contextualized representations learned by BERT to term weights, then uses predicted term weights to replace the original TF field in the inverted index. Experimental results show that predicted weights could better estimate the term importance and improve term-based methods for the first-stage retrieval. Moreover, results in the work of Mackenzie et al. [143] verify that DeepCT can improve search efficiency via a static index pruning technique. Furthermore, Dai and Callan [50] introduced the HDCT model to learn term weights for long documents. It first estimates passage-level term weights using contextual term representations produced by BERT. Then, passage-level term weights are combined into document-level term weights through a weighted sum. It is worth noting that the learned term weights by the preceding models, including DeepCT and HDCT, are in the range of 0-1. Then, they scale the real-valued predictions into a *tf*-like integer. In this way, these term weights can be directly integrated into the existing inverted index and can be implemented with existing retrieval models.

The preceding approaches rely on neural embeddings, which are learned within local or global contexts, to predict term weights directly. In addition, there are some works trying to estimate term weights by evaluating the matching score between each term and the whole document through a complex interaction network. For example, Mitra et al. [156] proposed to incorporate query term independence assumption into three state-of-the-art neural ranking models (BERT [55], Duet [153], and Conv-KNRM [52]), and the final relevance score of the document can be decomposed with respect to each query term. In this way, these neural ranking models can be used to predict the matching score of each term to the document, which can be pre-computed and indexed offline. Experimental results on a passage retrieval task show that this method exhibits significant improvement over classical term-based methods, with only a small degradation compared with original neural ranking models. Similarity, Mitra et al. [154] extended the Transformer-Kernel [92] architecture to the full retrieval setting by incorporating the query term independence assumption. First, they simplified the query encoder by getting rid of all Transformer layers and only considering non-contextualized embeddings for query terms. Second, instead of applying the aggregation function over the full interaction matrix, they applied it to each row of the matrix individually, which corresponds to an individual matching score between each query term and the whole document.

In addition to explicitly predicting term weights, another kind of method is to augment the document with additional terms using neural sequence-to-sequence (seq2seq) models. In this way, term weights of those elite terms can be promoted in the inverted index. In fact, this kind of method follows the idea of document expansion described in Section 4.2, yet it does the expansion with neural networks. For example, the doc2query [164] model trains a seq2seq model based on relevant query-document pairs. Then, the seq2seq model generates several queries for each document, and those synthetic queries are appended to the original document, forming the "expanded document." This expansion procedure is performed on every document in the corpus, and the expanded document collection is indexed as usual. Finally, it relies on a BM25 algorithm to retrieve relevant candidates. When combined with a re-ranking component, it achieves the state-of-the-art performance on MS MARCO [160] and TREC CAR [59] retrieval benchmarks. Later, the docTTTTTquery model [162] employs a stronger pre-trained model T5 [173] to generate queries and achieves large

gains compared with doc2query. Moreover, Yan et al. [219] proposed a **Unified Encoder-Decoder (UED)** network to enhance document expansion with the document ranking task. Experimental results on two large-scale datasets show that UED achieves a new state-of-the-art performance on both the MS MARCO passage retrieval task and TREC 2019 Deep Learning Track.

There are also some works trying to learn term weights as well as document expansion simultaneously in a unified framework [14, 70, 145]. For example, Bai et al. [14] proposed a novel framework, SparTerm, to build term-based sparse representations in the full vocabulary space. It takes the pre-trained language model to map the frequency-based BOW representation to a sparse term importance distribution in the whole vocabulary. In this way, it can simultaneously learn the weights of existing terms and expand new terms for the document. In addition, SparTerm constructs a gating controller to generate binary and sparse signals across the dimension of vocabulary size, ensuring the sparsity of final representations. As well, DeepImpact [145] leverages docTTTTTquery to enrich the document collection and then uses a contextualized language model to estimate the semantic importance of tokens in the document. In this way, it can produce a single-value representation for the original token and expanded token in each document.

5.1.2 Sparse Representation Learning. In contrast to weighting terms in the symbolic space, sparse representation learning methods focus on building sparse vectors for queries and documents, where representations are expected to capture semantic meanings of each input text. In this way, queries and documents are represented in the latent space. But different from topic models in Section 4.4, each dimension of the latent space learned by neural models has no clear concepts. Then, the learned sparse representations can be stored and searched with an inverted index efficiently, where each unit in the inverted index table corresponds to a "latent word" instead of a term.

Learning sparse embeddings can be traced back to semantic hashing [182], which employs deep auto-encoders for semantic modeling. It takes a multi-layer auto-encoder to learn distributed representations for documents. This model captures the document-term information, but it does not model the relevance relationship between queries and documents. Thus, it still cannot outperform classical term-based retrieval models, such as BM25 and QL. Zamani et al. [228] proposed a stand-alone neural ranking model to learn latent sparse representation for each query and document. Specifically, it first maps each n-gram in queries and documents to a low-dimensional dense vector to compress the information and learn the low-dimensional manifold of the data. Then, it learns a function to transform n-gram representations to high-dimensional sparse vectors. Finally, the dot product is used as the matching function to calculate the similarity between each query and document. This architecture learns latent sparse representations to better capture semantic relationships between query-document pairs, showing better performance over traditional term-based retrieval and several neural ranking models. But it uses n-gram as an encoding unit, which can only capture local dependencies and cannot adjust dynamically to the global context. Recently, Jang et al. [100] presented UHD-BERT, a novel sparse retrieval method empowered by extremely high dimensionality and controllable sparsity. They showed that the model outperforms previous sparse retrieval models significantly and delivers competitive performance compared to dense retrieval models.

To make interaction-focused models applicable for the first-stage retrieval, Ji et al. [103] proposed to use sparse representations to improve the efficiency of three interaction-focused neural algorithms (DRMM [82], KNRM [215], and Conv-KNRM [52]). The work investigates a locality sensitive hashing [53] approximation of three neural methods with fast histogram-based kernel calculation and term vector pre-computing for a runtime cache. Evaluation results show that the proposed method yields 4.12×, 80.54×, and 106.52× speedups for DRMM, KNRM, and Conv-KNRM, respectively, on the ClueWeb dataset.

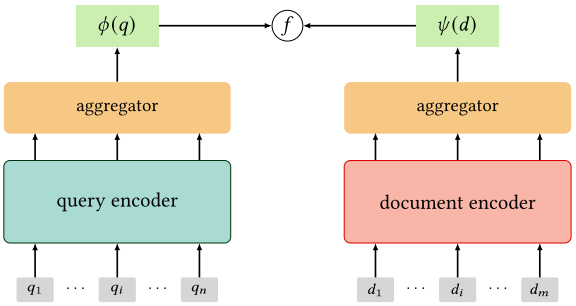


Fig. 2. Dual-encoder architecture of dense retrieval methods.

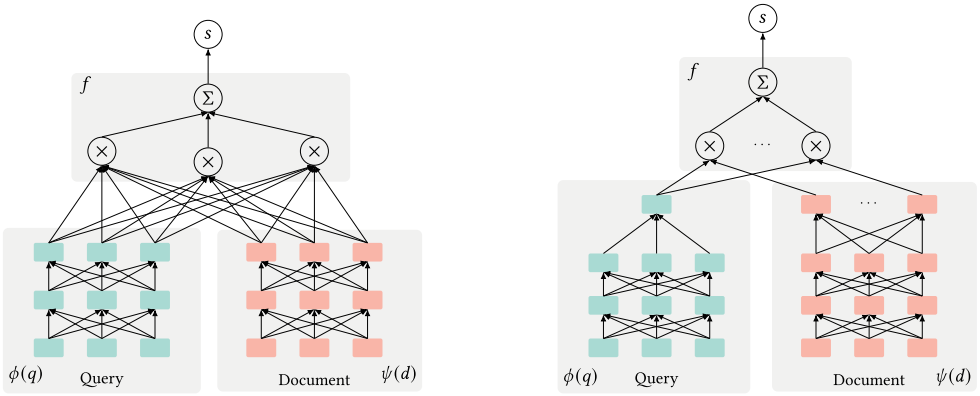
5.2 Dense Retrieval Methods

One of the biggest benefits of neural retrieval methods is to move away from sparse representations to dense representations, which is able to capture semantic meanings of input texts for better relevance evaluation. As shown in Figure 2, dense retrieval models usually have dual-encoder architecture, also called the *Siamese network* [30], which consists of twin networks that accept distinct inputs (queries and documents) and learn stand-alone dense embeddings for them independently. Then, the learned dense representations $\phi(q)$ and $\psi(d)$ are fed into a matching layer f, which is often implemented via a simple similarity function, to produce the final relevance score. To support the online serving, the learned dense representations are often indexed and searched via ANN algorithms [38, 106].

Researchers have devoted a lot of effort to designing sophisticated architectures to learn dense representations for retrieval. Due to the heterogeneous nature of text retrieval, the document often has abundant contents and complicated structures, so much attention has been paid to the design of the document-side representation function ψ. According to the form of learned document representations, we can divide dense retrieval models into two classes, as shown in Figure 3: *term-level representation learning* and *document-level representation learning*.

5.2.1 Term-Level Representation Learning. Term-level representation learning methods learn fine-grained term-level representations for queries and documents, and queries and documents are represented as a sequence/set of term embeddings. As is shown in Figure 3(a), the similarity function f then calculates term-level matching scores between the query and the document and aggregates them as the final relevance score.

One of the easiest methods is to take word embeddings, which have been proved to be effective in building ranking models for later re-ranking stages [82, 215], to build term-level representations for queries and documents. For example, Kenter and de Rijke [111] investigated whether it is possible to rely only on semantic features, such as word embeddings, rather than syntactic representations to calculate similarities between short texts. They replaced the $tf(q_i, d)$ in BM25 with the maximum cosine similarity between the word embedding of q_i and words in the document d. Their results show that the model can outperform baseline methods that work under the same condition. Mitra et al. [155] trained a Word2Vec embedding model on a large unlabeled query corpus, but in contrast to only retaining the output lookup table, they retained both input and output projections, allowing to leverage both embedding spaces to derive richer distributional relationships. During ranking, they mapped query words into the input space and document words into the output space, and computed the relevance score by aggregating cosine similarities across all query-document word pairs. The experimental results show that the DESM can re-rank top

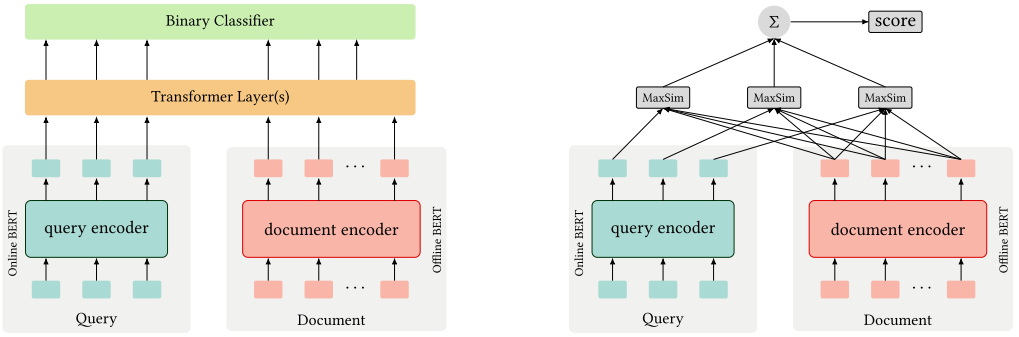


(a) Term-level representation learning (b) Document-level representation learning

Fig. 3. Different dense retrieval models for the first-stage retrieval.

documents returned by a commercial Web search engine like Bing better than a term-based signal like TF-IDF. However, when retrieving in a non-telescoping setting, DESM features are very susceptible to false-positive matches and can only be used either in conjunction with other document ranking features, such as TF-IDF, or for re-ranking a smaller set of candidate documents.

In recent years, the combination of contextual word embeddings and self-supervised pre-training has revolutionized the field of NLP and obtained state-of-the-art performance on many NLP tasks [55, 169, 221]. There are also a number of works that employ contextual word embeddings to learn query/document representations for IR. For example, Zhang et al. [237] proposed DC-BERT, which employs dual BERT encoders for low layers, as shown in Figure 4(a), where an online BERT encodes the query only once and an offline BERT pre-encodes all documents and caches all term representations. Then, the obtained contextual term representations are fed into high-layer Transformer interaction, which is initialized by the last k layers of the pre-trained BERT [55]. The number of Transformer layers K is configurable to a trade-off between the model capacity and efficiency. On the SQuAD dataset and Natural Questions dataset, DC-BERT achieves 10x speedup over the original BERT model on document retrieval while retaining most (about 98%) of the QA performance compared to state-of-the-art approaches for Open QA. An alternative way to use BERT for the term-level representation learning is the ColBERT [112] model, which employs a cheap yet powerful interaction function (i.e., a term-based MaxSim) to model fine-grained matching signals, as shown in Figure 4(b). Concretely, every query term embedding interacts with all document term embeddings via a MaxSim operator, which computes maximum similarity (e.g., cosine similarity or L2 distance), and scalar outputs of these operators are summed across query terms. Based on this, it can achieve cheap interaction and high-efficient pruning for top-k relevant documents retrieval. Results on MS MARCO and TREC CAR show that ColBERT's effectiveness is competitive with existing BERT-based models (and outperforms every non-BERT baseline), while executing two orders-of-magnitude faster and requiring four orders-of-magnitude fewer FLOPs per query. A similar model COIL is proposed by Gao et al. [77], but the query term embedding only interacts with exactly matched document term embeddings in the MaxSim operator. Experimental results show that COIL performs on par with more expensive and complex all-to-all matching retrievers (e.g., ColBERT). In addition, Cao et al. [34] and MacAvaney et al. [141] proposed DeFormer and PreTTR to decompose lower layers of BERT, which substitutes the full self-attention with question-wide and passage-wide self-attentions, as shown in Figure 5. The proposed approaches considerably



(a) DC-BERT (b) ColBERT

Fig. 4. Term-level representation learning methods. (a) The architecture of the DC-BERT [237] model. (b) The architecture of ColBERT [112].

reduce the query-time latency of deep Transformer networks. The difference between them is that the PreTTR model [141] inserts a compression layer to match attention scores to reduce the storage requirement up to 95% but without substantial degradation in retrieval performance.

A natural extension of the term-level representation learning is to learn phrase-level (i.e, n-grams, sentences) representations for documents, and documents are finally represented as a sequence/set of embeddings. Meanwhile, the query is usually viewed as one phrase and abstracted into a single vector, as it is often short in length. Then, the similarity function f calculates matching scores between the query with all phrases in the document and aggregates these local matching signals to produce the final relevance score. For example, Seo et al. [186] proposed to learn phrase representations based on BiLSTM for the OpenQA task. It leads to a significant scalability advantage since encodings of answer candidate phrases in the document can be pre-computed and indexed offline for efficient retrieval. Subsequently, Seo et al. [187] and Lee et al. [122] replaced the LSTM-based architecture with a BERT-based encoder, and augmented dense representations learned by BERT with contextualized sparse representations, improving the quality of each phrase embedding. Different from the document encoder, the query encoder only generates one embedding in capturing the whole contextual information of queries. Experimental results show that the OpenQA model that augments learned dense representations with learned contextual sparse representations outperforms previous OpenQA models, including recent BERT-based pipeline models, with two orders of magnitude faster inference time. For the multi-hop OpenQA task, Feldman and El-Yaniv [69] proposed the MUPPET model for efficient retrieval. The retrieval is performed by considering similarities between the question and contextualized sentence-level representations of the paragraph in the knowledge source. Given the sentence representations $(\boldsymbol{s}_1, \boldsymbol{s}_2, \ldots, \boldsymbol{s}_k)$ of a paragraph P, and the question encoding $\boldsymbol{q}$ for Q, the relevance score of P with respect to a question Q is calculated in the following way:

$$\mathrm{s}(Q,P) = \max_{i=1,\ldots,k} \sigma\left(\begin{bmatrix} \boldsymbol{s}_i \\ \boldsymbol{s}_i \odot \boldsymbol{q} \\ \boldsymbol{s}_i \cdot \boldsymbol{q} \\ \boldsymbol{q} \end{bmatrix} \cdot \begin{bmatrix} \boldsymbol{w}_1 \\ \boldsymbol{w}_2 \\ w_3 \\ \boldsymbol{w}_4 \end{bmatrix} + b\right), \tag{13}$$

where $\boldsymbol{w}_1, \boldsymbol{w}_2, \boldsymbol{w}_4 \in \mathbb{R}^d$ and $w_3, b \in \mathbb{R}$ are learned parameters. The method achieves state-of-the-art performance over two well-known datasets: SQuAD-Open and HotpotQA.

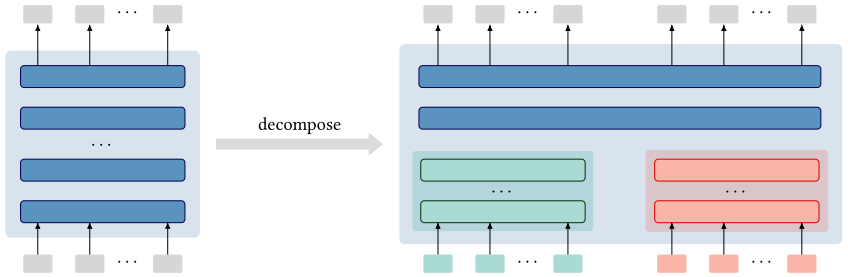


Fig. 5. Decompose BERT to question-wide and passage-wide self-attentions.

5.2.2 Document-Level Representation Learning. The document-level representation learning methods learn one or more coarse-level global representation(s) for each query and each document by abstracting their semantic meanings with dense vectors. It often employs a simple similarity function f (e.g., dot product, or cosine similarity) to calculate the final relevance score based on the query embedding $\phi(q)$ and the document embedding(s) $\psi(d)$, as is shown in Figure 3(b).

Initial attempts to obtain query embeddings and document embeddings are to directly aggregate their corresponding word embeddings with some pre-defined heuristic functions. Clinchant and Perronnin [45] were the first to propose a document representation model, **Fisher Vector (FV)**, based on continuous word embeddings. It first maps word embeddings into a higher-dimensional space, then aggregates them into a document-level representation through the Fisher kernel framework. Although the FV model outperforms LSI for ad hoc retrieval tasks, it does not perform better than classical IR models, such as TF-IDF and the divergence from randomness [7] retrieval model. Gillick et al. [81] proposed to utilize the average of word embeddings as the query or document representation. The experimental results show the proposed model outperforms term-based retrieval models (e.g., TF-IDF and BM25), which indicates that dense retrieval is a viable alternative to the discrete retrieval model. Obtaining text representations by aggregating word embeddings loses the contextual and word order information as classical term-based retrieval models do. To solve this problem, Le and Mikolov [120] proposed **Paragraph Vector (PV)**, an unsupervised algorithm that learns fixed-length representations from variable-length pieces of texts, such as sentences, paragraphs, and documents. Ai et al. [5, 6] evaluated the effectiveness of PV representations for ad hoc retrieval but produced unstable performance and limited improvements. With many attempts that use word/document embeddings to obtain dense representations for queries and documents, only moderate and local improvements over traditional term-based retrieval models have been observed, suggesting the need for more IR-customized embeddings or more powerful representation learning models.

As for embeddings customized for IR, Ai et al. [5] analyzed intrinsic problems of the original PV model that restrict its performance on retrieval tasks. Then, they produced modifications to the PV model, making it more suitable for IR tasks. The evaluation results on Robust04 and GOV2 show the effectiveness of the enhanced PV model. Subsequently, Gysel et al. [85] proposed the **neural vector space model (NVSM)**, an unsupervised method that learns latent representations of words and documents from scratch for news article retrieval. The query is represented by averaging its constituent word representations and projected to the document feature space. The matching score between a document and a query is given by the cosine similarity between their representations in document feature space. The experiments show that the NVSM outperforms lexical retrieval models on four article retrieval benchmarks. Similar to the NVSM, another unsupervised embedding learning method tailored for IR is SAFIR [4]. SAFIR jointly learns word,

concept, and document representations from scratch. The similarity of a query to a document is calculated by averaging its word-concept representations and then projecting it into the document space. Finally, the matching score between the query and the document is given by the cosine similarity between their representations in the document space. The evaluation on shared test collections for medical literature retrieval shows the effectiveness of SAFIR in terms of retrieving relevant documents. In addition to optimizing word/document embeddings for retrieval objectives directly, considering external knowledge resources, such as semantic graphs, ontologies and knowledge graphs, to enhance embeddings learning for semantic retrieval is another effective solution [136, 159, 197]. For example, Liu et al. [136] leveraged the existing knowledge (word relations) in the medical domain to constrain word embeddings using the principle that related words should have similar embeddings. The resulting constrained word embeddings are used for IR tasks, showing superior effectiveness to unsupervised word embeddings.

For more powerful representation learning models for the first-stage retrieval, Henderson et al. [87] proposed a computationally efficient neural method for natural language response suggestion. The feed-forward neural network uses n-gram embedding features to encode messages and suggested replies into vectors, which is optimized to give message-response pairs higher dot product values. The DPR [110] model is proposed to learn dense embeddings for text blocks with a BERT-based dual encoder. The retriever based on the DPR model outperforms a strong Lucene BM25 system on a wide range of OpenQA datasets and is beneficial for the end-to-end QA performance. Similar to DPR, the RepBERT [232] model employs a dual encoder based on BERT to obtain query and document representations, then inner products of query and document representations are regarded as relevance scores. Experimental results show that the RepBERT outperforms BM25 on the MS MARCO passage ranking task.

Another alternative approach is to distill a more complex model (e.g., a term-level representation learning method or interaction-focused model) to a document-level representation learning architecture. For example, Lin et al. [132] distilled the knowledge from ColBERT's expressive MaxSim operator for computing relevance scores into a simple dot product, thus enabling a single-step ANN search. Their key insight is that during distillation, tight coupling between the teacher model and the student model enables more flexible distillation strategies and yields better learned representations. The approach improves query latency and greatly reduces the onerous storage requirement of ColBERT while only making modest sacrifices in terms of effectiveness. Tahami et al. [196] utilized knowledge distillation to compress the complex BERT cross-encoder network as a teacher model into the student BERT bi-encoder model. This increases the prediction quality of BERT-based bi-encoders without affecting its inference speed. They evaluated the approach on three domain-popular datasets, and results show that the proposed method achieves statistically significant gains.

It should be noted that among neural models proposed early for IR tasks, such as DSSM [96], ARC-I [93], and QA_LSTM [198], they learn highly abstract document representations based on different network architectures, such as the fully connected network, **convolutional neural network (CNN)**, and RNN. Then a simple matching function, such as cosine similarity and bilinear, is used to evaluate similarity scores. These models are usually proposed for re-ranking stages at the beginning; however, because of their dual-encoder architecture, it is theoretically possible that they are also applicable for the first-stage retrieval. Nevertheless, a study by Guo et al. [82] shows that DSSM, C-DSSM [189], and ARC-I perform worse when trained on a whole document than when trained only on titles. Due to these limitations, most of these early neural models fail to beat unsupervised term-based retrieval baselines (e.g., BM25) on academic benchmarks. These drawbacks motivate the development of models discussed in this survey that are designed specifically for the retrieval stage.

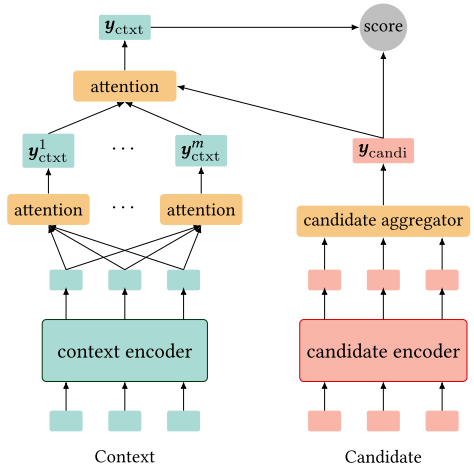


Fig. 6. Document-level multi-vector representation method in Poly-encoders [97].

In addition to learn a single global representation for each query and each document, another more sophisticated approach is to employ different encoders for queries and documents, where the document encoder abstracts the content into multiple embeddings—each embedding captures some aspects of the document, whereas the query encoder obtains a single embedding for each query [97, 137, 199]. The motivation is that documents are often lengthy and have diverse aspects in them, but queries are usually short and have focused topics. For example, Luan et al. [137] proposed Multi-Vector BERT (ME-BERT) to obtain a single-vector representation for the query and a multi-vector representation for the document. They represented the sequence of contextualized query/document embeddings at the top level of a deep Transformer, then defined the single-vector query representation as the contextualized embedding of the special token "[CLS]" and the multi-vector document representation as the first m contextualized vectors of tokens in the document. The value of m is always smaller than N, where N is the number of tokens in the document. Finally, the relevance score is calculated as the largest inner product yielded by each document vector with the query vector. Experimental results show that the ME-BERT model yields strong performance than alternatives in open retrieval. Similarly, Humeau et al. [97] proposed Poly-encoders, an architecture with an additional learned attention mechanism to represent more global features. Poly-encoders, as shown in Figure 6, uses two separate Transformer models to encode contexts and candidates. The candidate is encoded into a single vector $\boldsymbol{y}_{\text{candi}}$, and the input context, which usually includes more information than a candidate, is represented with m vectors $(\boldsymbol{y}^1_{\text{ctxt}}, \ldots, \boldsymbol{y}^m_{\text{ctxt}})$ instead of just one. Then, m vectors are attended using the candidate encoding vector $\boldsymbol{y}_{\text{candi}}$ to get the final score. The value of m will give a trade-off between inference accuracy and speed. It should be noted that different from general retrieval tasks that retrieved texts (documents) are usually longer than input texts (queries), the task in the work of Humeau et al. [97] has longer input texts than retrieved texts, and thus the multi-vector representation model is actually employed for the query encoder in their work [97].

5.3 Hybrid Retrieval Methods

Sparse retrieval methods take words or "latent words" as the unit of indexing, which preserves strong discriminative power as the score is calculated by hard matching between each unit. As a result, they can identify exact matching signals, which are momentous for retrieval tasks. However, dense retrieval methods learn continuous embeddings to encode semantic information and

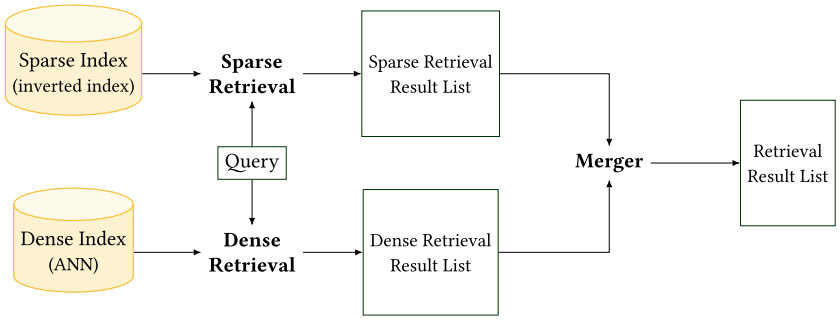


Fig. 7. The general architecture of hybrid retrieval methods.

soft matching signals, but detailed low-level features are always sacrificed. A natural approach to balance between the fidelity of sparse retrieval methods and the generalization of dense retrieval methods is to combine merits of them to build a hybrid retrieval model [78, 85, 115, 207]. Hybrid retrieval methods define multiple representation functions (ϕ and ψ), then obtain sparse and dense representations for queries/documents. Finally, these representations are used to calculate the final matching score with different merging ways (f). The general architecture of hybrid retrieval methods is shown in Figure 7.

With the development of the word embedding technique, there are a number of works on exploiting it with term-based models for the first-stage retrieval. Vulić and Moens [207] obtained better results on monolingual and bilingual retrieval by combining the word-embedding-based method with a uni-gram language model. However, the embedding-based model solely does not outperform traditional language models in the monolingual retrieval task. In other words, the effectiveness of neural semantic retrieval models is more observed when combined with term-based retrieval methods instead of replacing them. The consistent observation is also obtained in other works [155, 158], in which direct use of word embeddings only obtains extremely poor performance in the non-telescoping setting, unless combining it with a term-based feature, such as BM25. The GLM [73] is an embedding-based translation model linearly combined with a traditional language model. The probability of observing a term t in the query from a document d is modeled by three parts—that is, direct term sampling, generating a different term t' either from the document itself or from the collection, and then transforming it to the observed query term t. The empirical results show that the GLM performs better than the traditional language model. Roy et al. [180] also proposed to combine word vector based query likelihood with the standard language model based query likelihood for document retrieval. Experiments on standard text collections show that the combined similarity measure almost always outperforms the language model similarity measure significantly. In addition, according to the experimental results obtained in the work of Gysel et al. [85], although the NVSM outperforms term-based retrieval models on some benchmarks, it will be more useful as a supplementary signal to term-based models. Similar conclusions could also be found in other works [4, 136, 197].

Different from using word embeddings to construct dense representations and using TF to obtain term-based matching scores directly, there are also some works trying to employ simple neural networks to learn sparse and dense representations, then calculate matching scores based on learned representations. For example, dos Santos et al. [62] proposed the BOW-CNN architecture to retrieve similar questions in online QA community sites, as shown in Figure 8, which combines a BOW representation with a distributed vector representation created by a CNN. The BOW-CNN model computes two partial similarity scores: $s_{\text{bow}}(q_1, q_2)$ for BOW representations

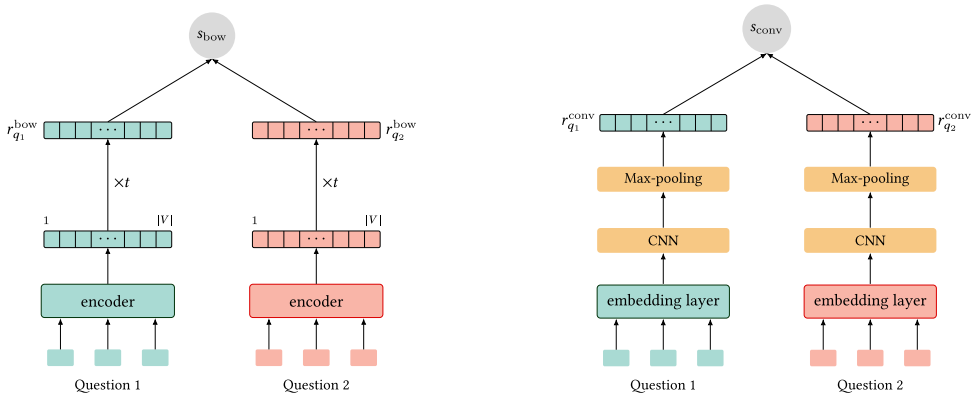


Fig. 8. The sparse and dense scoring component in BOW-CNN [62].

and $s_{\text{conv}}(q_1, q_2)$ for CNN representations. Finally, it combines two partial scores to create the final score $s(q_1, q_2)$. They performed experiments on two datasets collected from Stack Exchange communities. The experimental results evidence that BOW-CNN is more effective than BOW-based IR methods such as TF-IDF, and BOW-CNN is more robust than the pure CNN for long texts. In addition, MacAvaney et al. [142] proposed a new approach for passage retrieval, which trains a model to generate query and document representations in a given fixed-length vector space, and produce a ranking score by computing a similarity score between two representations. Different from other representation learning methods, it represents each query as a sparse vector and each document as a dense vector. Finally, the dot product is used to compute the similarity between the query vector and the document vector. The experimental results show that the proposed EPIC model significantly outperforms prior approaches. It is also observed that the performance is additive with current leading first-stage retrieval methods.

With the rise of more powerful pre-training neural networks (e.g., BERT, GPT-3), it is a natural way to combine them with term-based models for improving the first-stage retrieval. Seo et al. [187] proposed DenSPI for the retrieval stage of OpenQA. The DenseSPI model constructs the dense-sparse representation for each phrase unit. The dense vector is represented as pointers to the start and end BERT-based token representations of the phrase, which is responsible for encoding syntactic or semantic information of the phrase with respect to its context. The sparse embedding uses 2-gram-based TF-IDF for each phrase, which is good at encoding precise lexical information. Later, Lee et al. [122] proposed to learn contextual sparse representation for each phrase based on BERT to replace TF-based sparse encodings in DenSPI [187]. This method leverages rectified self-attention to indirectly learn sparse vectors in n-gram vocabulary space, improving the quality of each phrase embedding by augmenting it with a contextualized sparse representation. Experimental results show that the OpenQA model that augments DenSPI with learned contextual sparse representations outperforms previous OpenQA models, including recent BERT-based pipeline models, with two orders of magnitude faster inference time. Luan et al. [137] proposed to linearly combine the term-based system (BM25-uni) and neural-based system (dual-encoder or multi-vector model) scores using a single trainable weight λ, tuned on a development set, which yields strong performance while maintaining the scalability. Gao et al. [78] proposed the CLEAR model, which uses a BERT-based embedding model to complement the term-based model (BM25). Experimental results show that retrieval from CLEAR without re-ranking is already almost as accurate as the BERT re-ranking pipeline. Similarly, Kuzi et al. [115] proposed a general hybrid approach for document

retrieval that leverages both a semantic model (BERT) and a lexical retrieval model (BM25). An in-depth empirical analysis is performed, which demonstrates the effectiveness of the hybrid approach and also sheds some light on the complementary nature of the lexical and semantic models.

5.4 Model Learning

As described earlier, neural semantic retrieval models always define functions ϕ, ψ, and f in the network structure. These functions are usually learned from data using deep learning technology. Here, we discuss key topics on the learning of neural semantic retrieval models, including loss functions and negative sampling strategies.

5.4.1 Loss Functions. We review major training objectives adopted by neural semantic retrieval models. Ideally, after the training loss is minimized, all preference relationships between documents should be satisfied and the model will produce the optimal result list for each query. This makes training objectives effective in many tasks where performance is evaluated based on the ranking of relevant documents.

In practice, the most commonly used loss function is *sampled cross-entropy loss*, also called n*egative log likelihood loss*:

$$\mathcal{L}\left(q, d^+, D^-\right) = -\log \frac{\exp\left(s\left(q, d^+\right)\right)}{\exp\left(s\left(q, d^+\right)\right) + \sum_{d^- \in D^-} \exp\left(s\left(q, d^-\right)\right)}, \tag{14}$$

where q denotes a query, d^+ is a relevant document of q, and D^- is the irrelevant document set of q.

Another commonly used loss function is the *hinge loss*:

$$\mathcal{L}\left(q, d^+, D^-\right) = \frac{1}{n} \sum_{d^- \in D^-} \max\left(0, m - \left(s(q, d^+) - s(q, d^-)\right)\right), \tag{15}$$

where q denotes a query, d^+ is a relevant document of q, D^- is the irrelevant document set of q, n is the number of documents in D^-, and m is the margin that is usually set as 1.

In fact, the negative log likelihood loss (Equation (14)) and hinge loss (Equation (15)) are also widely used in many other tasks with different names, such as InfoNCE loss in contrastive representation learning [41, 201] and Bayesian personalized ranking loss in recommender systems [175]. These loss functions and their variations have been well studied in other fields, including extreme multi-class classification [15, 24, 174], representation learning [41, 201], and deep metric learning [194, 209, 210], among others. The research progress in these fields might provide some insights to inspire the loss design in neural semantic retrieval. First of all, Wang et al. [210] showed that the softmax log likelihood loss is actually a smooth version of hinge loss. Moreover, several works have shown that the concept of margin in hinge loss can also be introduced into softmax cross-entropy loss to improve the performance in tasks like face recognition and Person Re-identification [194, 209]. In addition, some works [41, 210, 220] in different domains all verify that applying the ℓ_2 normalization to final representations (i.e., using cosine as the score function f) along with temperature can make the learning robust and improve the performance. Another line of research focuses on the bias in sampled softmax cross-entropy loss [42, 101]. For example, works in NLP [24, 101] usually focus on the unbiased estimation of the full softmax, whereas Chuang et al. [42] focused on correcting the bias introduced by the false-negative samples that have the same label as the ground truth. It is worth noting that these conclusions need to be re-examined under the first-stage retrieval task.

5.4.2 Negative Sampling Strategies. In loss functions (Equation (14) and Equation (15)), the negative example set D^- is an important part of inputs. However, during the learning of the first-stage

retrieval models, it is often the case that only positive examples are available in the training dataset, whereas negative examples are not explicitly labeled. In fact, the sampling strategy of negative examples is a crucial topic in neural semantic retrieval models, because it directly determines the quality of the learned retrieval model.

Negative sampling is a common fundamental problem in the learning of many tasks where only positive signals are explicitly existed, like recommender systems [60, 220, 225, 226, 236], graph mining [9, 195, 222], and self-supervised representation learning [27, 37, 41, 86, 238]. Here, we mainly focus on the research progress in the field of the first-stage retrieval. The neural semantic retrieval models usually vary in their mechanisms to construct negative examples. But in general, negative sampling strategies can be divided into three categories:

(1) *Random negative sampling:* Random samples from the entire corpus [110, 137] or in batch [81, 87, 88, 232]. It should be noted that if using the batch as a source for random negatives, the batch size becomes important [81]. Lee et al. [123] suggested to use a large batch size because it makes the training task more difficult and closer to what the retriever observes at test time. However, the batch size is usually restricted by computing resources and cannot be set very largely. To address this problem, He et al. [86] proposed to decouple the size of mini-batch and sampled negative examples by maintaining a queue of data samples (encoded representations of the current mini-batch are enqueued, and the oldest are dequeued) to provide negative samples. In this way, they can use a very large size (e.g., 65,536) for negative samples in unsupervised visual representation learning. However, random negative sampling is usually sub-optimal for training neural semantic retrieval models. Models can hardly focus on improving top-ranking performance since these random negative samples are usually too easy to be distinguished. This problem would lead to serious performance dropping in practice. To make the model better at differentiating between similar results, one can use samples that are closer to positive examples in the embedding space as hard negatives for training. Thus, mining hard negative samples to optimize retrieval performance is a key problem that needs to be addressed.

(2) *Static hard negative sampling:* Random samples from pre-retrieved top documents by a traditional retriever [78, 110, 137], such as BM25. Recent researches find it helps training convergence to include BM25 negatives to provide stronger contrast for representations learning [110, 137]. Obtaining hard negative samples with pre-retrieval is computationally efficient. However, hard negative samples obtained by static methods are not real hard negatives. Intuitively, strong negatives close to relevant documents in an effective neural retrieval model space should be different from those from term-based retrieval models, as the goal of neural semantic retrieval models is to find documents beyond those retrieved by term-based models. If using negative samples from BM25, there exists a severe mismatch between negatives used to train the retrieval model and those seen in testing.

(3) *Dynamic hard negative sampling:* Random samples from top-ranked irrelevant documents predicted by the retrieval model itself. Intuitively, negative sampling dynamically according to current semantic retrieval models (e.g., using the distribution which is proportional to relevance scores predicted by the current model) should be a very promising choice for producing informational negative samples [166, 195]. In this way, neural semantic retrieval models can optimize themselves using negative samples they did wrong (i.e., predict a high relevance score for an irrelevant document). However, it is usually impractical to score all candidate documents in a very large corpus on the fly. Thus, in real-world settings, periodically refreshing the index and retrieving top-ranked documents as hard negatives is a more practical compromise choice [61, 78, 95, 216]. For example, hard negatives

mining in the work of Xiong et al. [216] elevates the BERT-based Siamese architecture to robustly exceed term-based methods for document retrieval. It also convincingly surpasses concurrent neural semantic retrieval models for passage retrieval on OpenQA benchmarks.

It should be noted that the negative sampling strategies described previously are not exclusive mutually. In practice, random sampled easy negatives and hard negatives are always used simultaneously. For example, the counterintuitive finding in the work of Huang et al. [95] shows that models trained simply using hard negatives cannot outperform models trained with random negatives. The hypothesis is that the presence of easy negatives in training data is still necessary, as a retrieval model is to operate on an input space that comprises data with mixed levels of hardness, and the majority of documents in the collection are easy cases that do not match the query at all. Having all negatives being so hard will change the representativeness of the training data to the real retrieval task, which might impose a non-trivial bias to learned embeddings.

Takeaway. Neural semantic retrieval methods learn the representation functions (i.e, ϕ and ψ) and the scoring function (f) with deep learning technologies. To support fast retrieval, document representations are often learned with stand-alone networks, and pre-computed and stored with delicate structures. According to how the representations are computed and stored, we summarize neural semantic retrieval methods into three paradigms—sparse retrieval methods, dense retrieval methods, and hybrid retrieval methods:

- Sparse retrieval methods focus on improving classical term-based methods by either learning to re-weight terms with contextual semantics or mapping texts into "latent word" space. Empirical results show that sparse retrieval methods could indeed improve the performance of the first-stage retrieval, and they are easily integrated with the existing inverted index for efficient retrieval. Moreover, these methods often show good interpretability as each dimension of the representation corresponds to a concrete token or a latent word.
- Dense retrieval methods employ the dual-encoder architecture to learn stand-alone low-dimensional dense vectors for queries and documents, aiming to capture the global semantics of input texts. To support online services, the learned dense representations are often indexed and searched via ANN algorithms. These methods have shown promising results on several benchmarks (e.g., MS MARCO and TREC CAR), and attracted increasing attention of researchers.
- Hybrid retrieval methods define multiple representation functions for queries and documents, then obtain their sparse and dense representations simultaneously for matching. They are able to achieve a balance between the fidelity of sparse retrieval methods and the generalization of dense retrieval methods. As a result, hybrid retrieval methods show better performance in practice but require much higher space occupation and retrieval complexity.

For neural semantic retrieval models learning, the negative sampling strategy is decisive for learning a high-quality retrieval model. Currently, there have been several works to explore better negative sampling methods, but it is still an open problem on how to mine negative documents for efficient and effective model learning.

6 CHALLENGES AND FUTURE DIRECTIONS

In this section, we discuss some open challenges and several future directions related to semantic models for the first-stage retrieval. Some of these topics are important but have not been well addressed in this field, whereas others are very promising directions for future research.

6.1 Pre-Training Objectives for the Retrieval Stage

Starting in 2018, there was rapid progress in different NLP tasks with the development of large pre-training models, such as BERT [55] and GPT [172]. They are pre-trained on the large-scale corpus and general-purpose modeling tasks such that the knowledge can be transferred into a variety of downstream tasks. With this intriguing property, one would expect to repeat these successes for IR tasks.

Some researchers [35, 84, 123] have explored pre-training models for the retrieval stage with a dual-encoder architecture. For example, Lee et al. [123] proposed to pre-train the two-tower Transformer encoder model with the **Inverse Cloze Task (ICT)** to replace BM25 in the passage retrieval stage for the OpenQA task. The advantage is that the retriever can be trained jointly with the reader. Nevertheless, the pre-training model does not outperform BM25 on the SQuAD dataset, potentially because the fine-tuning is only performed on the query-tower. Except for the ICT pre-training task, Chang et al. [35] also proposed the Body First Selection and Wiki Link Prediction tasks, and studied how various pre-training tasks help the large-scale retrieval problem, such as passage retrieval for OpenQA. The experimental results show that with properly designed paragraph-level pre-training tasks including ICT, Body First Selection, and Wiki Link Prediction, the two-tower Transformer encoder model can considerably improve over the widely used BM25 algorithm. In addition, Ma et al. [139, 140] proposed pre-training with the Representative Words Prediction task for ad hoc retrieval, which achieves significant improvement over baselines without pre-training or with other pre-training methods. However, whether the Representative Words Prediction task works for the retrieval stage needs to be re-examined since their experiments are conducted under re-ranking stages.

In summary, there has been little effort to design large pre-training models toward the first-stage retrieval task. As is known to all, the first-stage retrieval mainly focuses on the capability to recall as many potentially relevant documents as possible. Thus, considering retrieval requirements in recalling relevant documents and modeling task-dependent characteristics would be important elements during designing novel pre-training objectives for the retrieval stage. In addition, using cross-modal data (e.g., images) to enhance language understanding is also a promising direction in pre-training research.

6.2 More Effective Learning Strategies

For IR tasks, the construction of benchmark datasets often relies on a pooling process to recall a subset of documents for expert judging. Such a labeling process leads to the well-known bias problem, where the dataset only contains partially positive documents and the rest of the unlabeled documents are oftentimes assumed to be equally irrelevant [110, 232]. To address the bias problem, it is necessary to devise smart learning strategies to achieve effective and efficient model training. For example, Chuang et al. [42] developed a debiased contrastive objective that corrects for the sampling of the same label data points, even without knowledge of true labels. Next, as discussed in Section 5.4.2, hard negative samples can improve the model's ability to differentiate between similar examples. However, hard negatives mining strategies have not been fully explored. One of the state-of-the-art methods is the Asynchronous ANCE training proposed by Xiong et al. [216], which periodically refreshes the ANN index and samples top-ranked documents as negatives. Although ANCE is competitive in terms of effectiveness, refreshing the index periodically greatly increases the model training cost (e.g., 10 hours for each period). In addition, some works conclude that it would be more effective to learn semantic retrieval models with hard negative samples and easy negative samples simultaneously [231]. Thus, in addition to mining hard negatives, it is also worthy to explore arranging the position and order of training samples since

negative documents often show varied levels of difficulty. We believe it would be interesting and valuable to study more complex training strategies, such as curriculum learning [18], to help the model optimization for the first-stage retrieval. Moreover, the supervised data for IR is always scarce since it requires much manual labor to obtain. In addition, the supervised dataset is prone to long-tail, sparsity, and other issues. Thus, weak supervised and unsupervised learning (e.g., contrastive learning [41, 86]) are promising directions. For example, Dai and Callan [50] proposed a content-based weak supervision strategy that exploits the internal structure of documents to mine training labels.

6.3 Benchmark Testbed for Efficiency Comparison

The multi-stage retrieval paradigm aims to balance between the effectiveness and efficiency of retrieval tasks, where the first-stage retrieval focus on the efficiency and re-ranking stages pay more attention to the effectiveness. But efficiency metrics in isolation are meaningless unless contextualized with corresponding effectiveness measures. Ideally, the efficiency metrics at different effectiveness cutoffs should be reported on the leaderboard. Moreover, since the customized hardware (e.g., GPUs or TPUs) has a significant impact on the computation time of deep models, and the response time of the first-stage retrieval models is also infamously sensitive to constraints, such as locality of data on file systems for caching, it is expected to compare different models under the same conditions. However, fair conditions for model efficiency comparison have not been fully valued and studied in the IR field as in the computer vision community. For example, the medical computer vision community has already recognized the need for a focus on runtime considerations. The medical image analysis benchmark VISCERAL [104] includes runtime measurements of participant solutions on the same hardware. Additionally, computer vision tasks, such as object detection and tracking, often require real-time results [94]. For IR tasks, Hofstätter and Hanbury [91] put forward a preliminary solution, which makes the comparison of runtime metrics feasible by introducing docker-based submissions of complete retrieval systems so that all systems can be compared under the same hardware conditions by a third party.

6.4 Advanced Indexing Schemes

As described in Section 3.2, for IR tasks, indexing schemes play an important role in determining the way to organize and retrieve large-scale documents. Specially, most dense retrieval methods, which learn dense representations for queries and documents, rely on ANN algorithms to perform efficient vector search for online services [32, 112].

Existing dense retrieval methods always separate two steps of representation learning and index building. This pattern suffers from a few drawbacks in practical scenarios. First, the indexing process cannot benefit from supervised information because it uses the task-independent function to build the index. In addition, the representation and index are separately obtained and thus may not be optimally compatible. These problems all result in severely decayed retrieval performance. In fact, there have been studies [227, 233] to explore the joint training of encoders and indexes in the fields of image retrieval and recommendation. For IR, it is still in its infancy stage to design joint learning schemes of the first-stage retrieval models and indexing methods, and we believe it would be an interesting and promising direction.

However, how to design better ANN algorithms that can manage large-scale documents and support efficient and precise retrieval is another important direction. Compared with the brute-force search, the essence of ANN search is to sacrifice part of precision to get higher retrieval efficiency. Generally, there are two kinds of ANN algorithms from the principle of improving retrieval efficiency. One is non-exhaustive ANN search methods [22, 144], and the other is vector compression methods [79, 99, 102]. However, each method has its limitations or deficiency, where

the non-exhaustive method has a large index size and the compression method has suboptimal performance. Thus, with the booming development of dense retrieval methods, it is urgent to develop more advanced ANN search algorithms to achieve a better balance between the efficiency and effectiveness.

7 CONCLUSION

The purpose of this survey is to summarize the current research status on semantic retrieval models, analyze existing methodologies, and gain some insights for future development. It includes a brief review of early semantic retrieval methods, a detailed description of recent neural semantic retrieval methods and the connection between them. Specially, we pay attention to neural semantic retrieval methods and review them from three major paradigms: sparse retrieval methods, dense retrieval methods, and hybrid retrieval methods. We also refer to key topics about neural semantic retrieval models learning, such as loss functions and negative sampling strategies. In addition, we discuss several challenges and promising directions that are important for future research. We look forward to working with the community on these issues.

It is our hope that this survey can help researchers who are interested in this direction and will motivate new ideas by looking at past successes and failures. Semantic retrieval models are part of the broader research field of neural IR, which is a joint domain of deep learning and IR technologies with many opportunities for new research and applications. We are expecting that, through the effort of the community, significant breakthroughs will be achieved for the first-stage retrieval problem in the near future, similar to those that have happened in re-ranking stages.

REFERENCES

[1] Mohammad Reza Abbasifard, Bijan Ghahremani, and Hassan Naderi. 2014. A survey on nearest neighbor search methods. *International Journal of Computer Applications* 95, 25 (2014), 39–52.

[2] Nasreen Abdul-Jaleel, James Allan, W. Bruce Croft, Fernando Diaz, Leah Larkey, Xiaoyan Li, Mark D. Smucker, and Courtney Wade. 2004. UMass at TREC 2004: Novelty and HARD. *Computer Science Department Faculty Publication Series* 2004 (2004), 189.

[3] Eneko Agirre, Xabier Arregi, and Arantxa Otegi. 2010. Document expansion based on WordNet for robust IR. In *Proceedings of COLING 2010: Posters*. 9–17.

[4] Maristella Agosti, Stefano Marchesin, and Gianmaria Silvello. 2020. Learning unsupervised knowledge-enhanced representations to reduce the semantic gap in information retrieval. *ACM Transactions on Information Systems* 38, 4 (2020), 1–48.

[5] Qingyao Ai, Liu Yang, Jiafeng Guo, and W. Bruce Croft. 2016. Analysis of the paragraph vector model for information retrieval. In *Proceedings of the 2016 ACM International Conference on the Theory of Information Retrieval*. 133–142.

[6] Qingyao Ai, Liu Yang, Jiafeng Guo, and W. Bruce Croft. 2016. Improving language estimation with the paragraph vector model for ad-hoc retrieval. In *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 869–872.

[7] Gianni Amati and Cornelis Joost Van Rijsbergen. 2002. Probabilistic models of information retrieval based on measuring the divergence from randomness. *ACM Transactions on Information Systems* 20, 4 (Oct. 2002), 357–389. https://doi.org/10.1145/582415.582416

[8] Alexandr Andoni. 2009. *Nearest Neighbor Search: The Old, the New, and the Impossible*. Ph.D. Dissertation. Massachusetts Institute of Technology, Cambridge, MA.

[9] Mohammadreza Armandpour, Patrick Ding, Jianhua Huang, and Xia Hu. 2019. Robust negative sampling for network embedding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 3191–3198. https://doi.org/10.1609/aaai.v33i01.33013191

[10] Avinash Atreya and Charles Elkan. 2011. Latent semantic indexing (LSI) fails for TREC collections. *ACM SIGKDD Explorations Newsletter* 12, 2 (2011), 5–10.

[11] Martin Aumüller, Erik Bernhardsson, and Alexander Faithfull. 2017. ANN-benchmarks: A benchmarking tool for approximate nearest neighbor algorithms. In *Proceedings of the International Conference on Similarity Search and Applications*. 34–49.

[12] Ricardo Baeza-Yates and Berthier Ribeiro-Neto. 2011. *Modern Information Retrieval: The Concepts and Technology Behind Search* (2nd ed.). Addison-Wesley.

[13] Jing Bai, Jian-Yun Nie, Guihong Cao, and Hugues Bouchard. 2007. Using query contexts in information retrieval. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'07)*. ACM, New York, NY, 15–22. https://doi.org/10.1145/1277741.1277747

[14] Yang Bai, Xiaoguang Li, Gang Wang, Chaoliang Zhang, Lifeng Shang, Jun Xu, Zhaowei Wang, Fangshan Wang, and Qun Liu. 2020. SparTerm: Learning term-based sparse representation for fast text retrieval. *arXiv preprint arXiv:2010.00768* (2020).

[15] Robert Bamler and Stephan Mandt. 2020. Extreme classification via adversarial softmax approximation. In *Proceedings of the International Conference on Learning Representations*. https://openreview.net/forum?id=rJxe3xSYDS.

[16] Petr Baudiš and Jan Šedivý. 2015. Modeling of the question answering task in the YodaQA system. In *Experimental IR Meets Multilinguality, Multimodality, and Interaction*, Josanne Mothe, Jacques Savoy, Jaap Kamps, Karen Pinel-Sauvagnat, Gareth Jones, Eric San Juan, Linda Capellato, and Nicola Ferro (Eds.). Springer International Publishing, Cham, Switzerland, 222–228.

[17] Jeffrey S. Beis and David G. Lowe. 1997. Shape indexing using approximate nearest-neighbour search in high-dimensional spaces. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE, Los Alamitos, CA, 1000–1006.

[18] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*. 41–48.

[19] Jon Louis Bentley. 1975. Multidimensional binary search trees used for associative searching. *Communications of the ACM* 18, 9 (1975), 509–517.

[20] Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. 2013. Semantic parsing on freebase from question-answer pairs. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*. 1533–1544. https://www.aclweb.org/anthology/D13-1160.

[21] Adam Berger and John Lafferty. 1999. Information retrieval as statistical translation. In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'99)*. ACM, New York, NY, 222–229. https://doi.org/10.1145/312624.312681

[22] Erik Bernhardsson. 2018. Annoy: Approximate nearest neighbors in C++/Python. *Python Package Version* 1, 0 (2018).

[23] Bodo Billerbeck and Justin Zobel. 2005. Document expansion versus query expansion for ad-hoc retrieval. In *Proceedings of the 10th Australasian Document Computing Symposium*. 34–41.

[24] Guy Blanc and Steffen Rendle. 2018. Adaptive sampled softmax with kernel based sampling. In *Proceedings of the 35th International Conference on Machine Learning*. 590–599.

[25] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent Dirichlet allocation. *Journal of Machine Learning Research* 3 (March 2003), 993–1022.

[26] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics* 5 (2017), 135–146. https://doi.org/10.1162/tacl_a_00051

[27] Avishek Joey Bose, Huan Ling, and Yanshuai Cao. 2018. Adversarial contrastive estimation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 1021–1032. https://doi.org/10.18653/v1/P18-1094

[28] Leonid Boytsov, David Novak, Yury Malkov, and Eric Nyberg. 2016. Off the beaten path: Let's replace term-based retrieval with k-NN search. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management (CIKM'16)*. ACM, New York, NY, 1099–1108. https://doi.org/10.1145/2983323.2983815

[29] Felipe Bravo-Marquez, Gaston L'Huillier, Sebastián A. Ríos, and Juan D. Velásquez. 2010. Hypergeometric language model and zipf-like scoring function for web document similarity retrieval. In *Proceedings of the 17th International Conference on String Processing and Information Retrieval (SPIRE'10)*. 303–308.

[30] Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Säckinger, and Roopak Shah. 1994. Signature verification using a "Siamese" time delay neural network. In *Advances in Neural Information Processing Systems 6*, J. D. Cowan, G. Tesauro, and J. Alspector (Eds.). Morgan-Kaufmann, 737–744. http://papers.nips.cc/paper/769-signature-verification-using-a-siamese-time-delay-neural-network.pdf.

[31] Robin D. Burke, Kristian J. Hammond, Vladimir Kulyukin, Steven L. Lytinen, Noriko Tomuro, and Scott Schoenberg. 1997. Question answering from frequently asked question files: Experiences with the FAQ FINDER system. *AI Magazine* 18, 2 (1997), 57.

[32] Yinqiong Cai, Yixing Fan, Jiafeng Guo, Ruqing Zhang, Yanyan Lan, and Xueqi Cheng. 2021. A discriminative semantic ranker for question retrieval. *arXiv preprint arXiv:2107.08345* (2021).

[33] Guihong Cao, Jian-Yun Nie, Jianfeng Gao, and Stephen Robertson. 2008. Selecting good expansion terms for pseudo-relevance feedback. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. 243–250.

[34] Qingqing Cao, Harsh Trivedi, Aruna Balasubramanian, and Niranjan Balasubramanian. 2020. DeFormer: Decomposing pre-trained transformers for faster question answering. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. 4487–4497. https://doi.org/10.18653/v1/2020.acl-main.411

[35] Wei-Cheng Chang, Felix X. Yu, Yin-Wen Chang, Yiming Yang, and Sanjiv Kumar. 2020. Pre-training tasks for embedding-based large-scale retrieval. In *Proceedings of the International Conference on Learning Representations*. https://openreview.net/forum?id=rkg-mA4FDr.

[36] Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading Wikipedia to answer open-domain questions. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 1870–1879. https://doi.org/10.18653/v1/P17-1171

[37] Long Chen, Fajie Yuan, Joemon M. Jose, and Weinan Zhang. 2018. Improving negative sampling for word representation using self-embedded features. In *Proceedings of the 11th ACM International Conference on Web Search and Data Mining (WSDM'18)*. ACM, New York, NY, 99–107. https://doi.org/10.1145/3159652.3159695

[38] Qi Chen, Haidong Wang, Mingqin Li, Gang Ren, Scarlett Li, Jeffery Zhu, Jason Li, Chuanjie Liu, Lintao Zhang, and Jingdong Wang. 2018. *SPTAG: A Library for Fast Approximate Nearest Neighbor Search*. GitHub. https://github.com/Microsoft/SPTAG.

[39] Qi Chen, Haidong Wang, Mingqin Li, Gang Ren, Scarlett Li, Jeffery Zhu, Jason Li, Chuanjie Liu, Lintao Zhang, and Jingdong Wang. 2018. *SPTAG: A Library for Fast Approximate Nearest Neighbor Search*. GitHub. https://github.com/Microsoft/SPTAG.

[40] Ruey-Cheng Chen, Luke Gallagher, Roi Blanco, and J. Shane Culpepper. 2017. Efficient cost-aware cascade ranking in multi-stage retrieval. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'17)*. ACM, New York, NY, 445–454. https://doi.org/10.1145/3077136.3080819

[41] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709* (2020).

[42] Ching-Yao Chuang, Joshua Robinson, Lin Yen-Chen, Antonio Torralba, and Stefanie Jegelka. 2020. Debiased contrastive learning. In *Advances in Neural Information Processing Systems*.

[43] Charles L. A. Clarke, Nick Craswell, and Ian Soboroff. 2004. Overview of the TREC 2004 terabyte track. In *Proceedings of the 13th Text REtrieval Conference (TREC'04)*, Vol. 4. 74.

[44] Charles L. Clarke, Nick Craswell, and Ian Soboroff. 2009. *Overview of the TREC 2009 Web Track*. Technical Report. Waterloo University, Ontario, Canada.

[45] Stéphane Clinchant and Florent Perronnin. 2013. Aggregating continuous word embeddings for information retrieval. In *Proceedings of the Workshop on Continuous Vector Space Models and Their Compositionality*. 100–109.

[46] Kevyn Collins-Thompson. 2009. Reducing the risk of query expansion via robust constrained optimization. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management (CIKM'09)*. ACM, New York, NY, 837–846. https://doi.org/10.1145/1645953.1646059

[47] Nick Craswell, Bhaskar Mitra, Emine Yilmaz, Daniel Campos, and Ellen M. Voorhees. 2020. Overview of the TREC 2019 Deep Learning Track. *arXiv preprint arXiv:2003.07820* (2020).

[48] Yiming Cui, Zhipeng Chen, Si Wei, Shijin Wang, Ting Liu, and Guoping Hu. 2017. Attention-over-attention neural networks for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 593–602. https://doi.org/10.18653/v1/P17-1055

[49] Zhuyun Dai and Jamie Callan. 2019. Context-aware sentence/passage term importance estimation for first stage retrieval. *arXiv preprint arXiv:1910.10687* (2019).

[50] Zhuyun Dai and Jamie Callan. 2020. Context-aware document term weighting for ad-hoc search. In *Proceedings of the Web Conference 2020 (WWW'20)*. ACM, New York, NY, 1897–1907. https://doi.org/10.1145/3366423.3380258

[51] Zhuyun Dai and Jamie Callan. 2020. Context-aware term weighting for first stage passage retrieval. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'20)*. ACM, New York, NY, 1533–1536. https://doi.org/10.1145/3397271.3401204

[52] Zhuyun Dai, Chenyan Xiong, Jamie Callan, and Zhiyuan 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 (WSDM'18)*. ACM, New York, NY, 126–134. https://doi.org/10.1145/3159652.3159659

[53] Mayur Datar, Nicole Immorlica, Piotr Indyk, and Vahab S. Mirrokni. 2004. Locality-sensitive hashing scheme based on p-stable distributions. In *Proceedings of the 20th Annual Symposium on Computational Geometry*. 253–262.

[54] Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard Harshman. 1990. Indexing by latent semantic analysis. *Journal of the American Society for Information Science* 41, 6 (1990), 391–407.

[55] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. 4171–4186. https://doi.org/10.18653/v1/N19-1423

[56] Bhuwan Dhingra, Kathryn Mazaitis, and William W. Cohen. 2017. Quasar: Datasets for question answering by search and reading. *arXiv preprint arXiv:1707.03904* (2017).

[57] Fernando Diaz. 2005. Regularizing ad hoc retrieval scores. In *Proceedings of the 14th ACM International Conference on Information and Knowledge Management (CIKM'05)*. ACM, New York, NY, 672–679. https://doi.org/10.1145/1099554.1099722

[58] Fernando Diaz, Bhaskar Mitra, and Nick Craswell. 2016. Query expansion with locally-trained word embeddings. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 367–377. https://doi.org/10.18653/v1/P16-1035

[59] Laura Dietz, Manisha Verma, Filip Radlinski, and Nick Craswell. 2017. TREC complex answer retrieval overview. In *Proceedings of the Twenty-Sixth Text REtrieval Conference, TREC (NIST Special Publication, Vol. 500-324)*, Ellen M. Voorhees and Angela Ellis (Eds.). National Institute of Standards and Technology (NIST), 1–13. https://trec.nist.gov/pubs/trec26/papers/Overview-CAR.pdf.

[60] Jingtao Ding, Yuhan Quan, Xiangnan He, Yong Li, and Depeng Jin. 2019. Reinforced negative sampling for recommendation with exposure data. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence (IJCAI'19)*. 2230–2236. https://doi.org/10.24963/ijcai.2019/309

[61] Yingqi Ding, Yuchen Qu, Jing Liu, Kai Liu, Ruiyang Ren, Xin Zhao, Daxiang Dong, Hua Wu, and Haifeng Wang. 2020. RocketQA: An optimized training approach to dense passage retrieval for open-domain question answering. *arXiv preprint arXiv:2010.08191* (2020).

[62] Cícero dos Santos, Luciano Barbosa, Dasha Bogdanova, and Bianca Zadrozny. 2015. Learning hybrid representations to retrieve semantically equivalent questions. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*. 694–699. https://doi.org/10.3115/v1/P15-2114

[63] Matthew Dunn, Levent Sagun, Mike Higgins, V. Ugur Güney, Volkan Cirik, and Kyunghyun Cho. 2017. SearchQA: A new Q&A dataset augmented with context from a search engine. *CoRR* abs/1704.05179 (2017). http://arxiv.org/abs/1704.05179.

[64] Karima Echihabi, Kostas Zoumpatianos, Themis Palpanas, and Houda Benbrahim. 2020. Return of the Lernaean Hydra: Experimental evaluation of data series approximate similarity search. *arXiv preprint arXiv:2006.11459* (2020).

[65] Miles Efron, Peter Organisciak, and Katrina Fenlon. 2012. Improving retrieval of short texts through document expansion. In *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 911–920.

[66] Anthony Fader, Luke Zettlemoyer, and Oren Etzioni. 2013. Paraphrase-driven learning for open question answering. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 1608–1618. https://www.aclweb.org/anthology/P13-1158.

[67] Joel L. Fagan. 1987. *Experiments in Automatic Phrase Indexing For Document Retrieval: A Comparison of Syntactic and Non-Syntactic Methods*. Technical Report. Cornell University.

[68] Miao Fan, Jiacheng Guo, Shuai Zhu, Shuo Miao, Mingming Sun, and Ping Li. 2019. MOBIUS: Towards the next generation of query-ad matching in Baidu's sponsored search. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'19)*. ACM, New York, NY, 2509–2517. https://doi.org/10.1145/3292500.3330651

[69] Yair Feldman and Ran El-Yaniv. 2019. Multi-hop paragraph retrieval for open-domain question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. 2296–2309. https://doi.org/10.18653/v1/P19-1222

[70] Thibault Formal, Benjamin Piwowarski, and Stéphane 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.

[71] Jibril Frej, Philippe Mulhem, Didier Schwab, and Jean-Pierre Chevallet. 2020. Learning term discrimination. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, New York, NY, 1993–1996. https://doi.org/10.1145/3397271.3401211

[72] George W. Furnas, Thomas K. Landauer, Louis M. Gomez, and Susan T. Dumais. 1987. The vocabulary problem in human-system communication. *Communications of the ACM* 30, 11 (1987), 964–971.

[73] Debasis Ganguly, Dwaipayan Roy, Mandar Mitra, and Gareth J. F. Jones. 2015. Word embedding based generalized language model for information retrieval. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'15)*. ACM, New York, NY, 795–798. https://doi.org/10.1145/2766462.2767780

[74] Jianfeng Gao, Xiaodong He, and Jian-Yun Nie. 2010. Clickthrough-based translation models for web search: From word models to phrase models. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management (CIKM'10)*. ACM, New York, NY, 1139–1148. https://doi.org/10.1145/1871437.1871582

[75] Jianfeng Gao and Jian-Yun Nie. 2012. Towards concept-based translation models using search logs for query expansion. *Microsoft*. Retrieved November 1, 2021 from https://www.microsoft.com/en-us/research/publication/towards-concept-based-translation-models-using-search-logs-query-expansion/.

[76] Jianfeng Gao, Jian-Yun Nie, Guangyuan Wu, and Guihong Cao. 2004. Dependence language model for information retrieval. In *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'04)*. ACM, New York, NY, 170–177. https://doi.org/10.1145/1008992.1009024

[77] Luyu Gao, Zhuyun Dai, and Jamie Callan. 2021. COIL: Revisit exact lexical match in information retrieval with contextualized inverted list. *arXiv preprint arXiv:2104.07186* (2021).

[78] Luyu Gao, Zhuyun Dai, Zhen Fan, and Jamie Callan. 2020. Complementing lexical retrieval with semantic residual embedding. *arXiv preprint arXiv:2004.13969* (2020).

[79] Tiezheng Ge, Kaiming He, Qifa Ke, and Jian Sun. 2013. Optimized product quantization. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 36, 4 (2013), 744–755.

[80] Daniel Gillick, Sayali Kulkarni, Larry Lansing, Alessandro Presta, Jason Baldridge, Eugene Ie, and Diego Garcia-Olano. 2019. Learning dense representations for entity retrieval. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL'19)*. 528–537. https://doi.org/10.18653/v1/K19-1049

[81] Daniel Gillick, Alessandro Presta, and Gaurav Singh Tomar. 2018. End-to-end retrieval in continuous space. *arXiv preprint arXiv:1811.08008* (2018).

[82] Jiafeng Guo, Yixing Fan, Qingyao Ai, and W. Bruce Croft. 2016. A deep relevance matching model for ad-hoc retrieval. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management (CIKM'16)*. ACM, New York, NY, 55–64. https://doi.org/10.1145/2983323.2983769

[83] Jiafeng Guo, Yixing Fan, Liang Pang, Liu Yang, Qingyao Ai, Hamed Zamani, Chen Wu, W. Bruce Croft, and Xueqi Cheng. 2019. A deep look into neural ranking models for information retrieval. *Information Processing & Management* 2019 (2019), 102067.

[84] Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. 2020. Realm: Retrieval-augmented language model pre-training. In *Proceedings of the 37th International Conference on Machine Learning*.

[85] Christophe Van Gysel, Maarten De Rijke, and Evangelos Kanoulas. 2018. Neural vector spaces for unsupervised information retrieval. *ACM Transactions on Information Systems* 36, 4 (2018), 1–25.

[86] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross B. Girshick. 2020. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR'20)*. IEEE. Los Alamitos, CA, 9726–9735. https://doi.org/10.1109/CVPR42600.2020.00975

[87] Matthew Henderson, Rami Al-Rfou, Brian Strope, Yun-Hsuan Sung, László Lukács, Ruiqi Guo, Sanjiv Kumar, Balint Miklos, and Ray Kurzweil. 2017. Efficient natural language response suggestion for smart reply. *arXiv preprint arXiv:1705.00652* (2017).

[88] Matthew Henderson, Ivan Vulić, Daniela Gerz, Iñigo Casanueva, Paweł Budzianowski, Sam Coope, Georgios Spithourakis, Tsung-Hsien Wen, Nikola Mrkšić, and Pei-Hao Su. 2019. Training neural response selection for task-oriented dialogue systems. *arXiv preprint arXiv:1906.01543* (2019).

[89] Djoerd Hiemstra. 2000. A probabilistic justification for using tf× idf term weighting in information retrieval. *International Journal on Digital Libraries* 3, 2 (2000), 131–139.

[90] Thomas Hofmann. 2017. Probabilistic latent semantic indexing. *SIGIR Forum* 51, 2 (Aug. 2017), 211–218. https://doi.org/10.1145/3130348.3130370

[91] Sebastian Hofstätter and Allan Hanbury. 2019. Let's measure run time! Extending the IR replicability infrastructure to include performance aspects. *arXiv preprint arXiv:1907.04614* (2019).

[92] Sebastian Hofstätter, Markus Zlabinger, and Allan Hanbury. 2020. Interpretable and time-budget-constrained contextualization for re-ranking. *arXiv preprint arXiv:2002.01854* (2020).

[93] Baotian Hu, Zhengdong Lu, Hang Li, and Qingcai Chen. 2014. Convolutional neural network architectures for matching natural language sentences. In *Advances in Neural Information Processing Systems 27*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger (Eds.). Curran Associates, 2042–2050. http://papers.nips.cc/paper/5550-convolutional-neural-network-architectures-for-matching-natural-language-sentences.pdf.

[94] Jonathan Huang, Vivek Rathod, Chen Sun, Menglong Zhu, Anoop Korattikara, Alireza Fathi, Ian Fischer, et al. 2017. Speed/accuracy trade-offs for modern convolutional object detectors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 7310–7311.

[95] Jui-Ting Huang, Ashish Sharma, Shuying Sun, Li Xia, David Zhang, Philip Pronin, Janani Padmanabhan, Giuseppe Ottaviano, and Linjun Yang. 2020. Embedding-based retrieval in Facebook search. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'20)*. ACM, New York, NY, 2553–2561. https://doi.org/10.1145/3394486.3403305

[96] Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry Heck. 2013. Learning deep structured semantic models for web search using clickthrough data. In *Proceedings of the 22nd ACM International Conference on Information and Knowledge Management (CIKM'13)*. ACM, New York, NY, 2333–2338. https://doi.org/10.1145/2505515.2505665

[97] Samuel Humeau, Kurt Shuster, Marie-Anne Lachaux, and Jason Weston. 2019. Poly-encoders: Transformer architectures and pre-training strategies for fast and accurate multi-sentence scoring. In *Proceedings of the International Conference on Learning Representations*. https://openreview.net/forum?id=SkxgnnNFvH.

[98] Prabhas Hundi and Rouzbeh Shahsavari. 2019. Deep learning to speed up the development of structure–property relations for hexagonal boron nitride and graphene. *Small* 15, 19 (2019), 1900656.

[99] Piotr Indyk and Rajeev Motwani. 1998. Approximate nearest neighbors: Towards removing the curse of dimensionality. In *Proceedings of the 30th Annual ACM Symposium on Theory of Computing*. 604–613.

[100] Kyoung-Rok Jang, Junmo Kang, Giwon Hong, Sung-Hyon Myaeng, Joohee Park, Taewon Yoon, and Heecheol Seo. 2021. UHD-BERT: Bucketed ultra-high dimensional sparse representations for full ranking. *arXiv preprint arXiv:2104.07198* (2021).

[101] Sébastien Jean, Kyunghyun Cho, Roland Memisevic, and Yoshua Bengio. 2015. On using very large target vocabulary for neural machine translation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. 1–10. https://doi.org/10.3115/v1/P15-1001

[102] Herve Jegou, Matthijs Douze, and Cordelia Schmid. 2010. Product quantization for nearest neighbor search. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 33, 1 (2010), 117–128.

[103] Shiyu Ji, Jinjin Shao, and Tao Yang. 2019. Efficient interaction-based neural ranking with locality sensitive hashing. In *Proceedings of the World Wide Web Conference (WWW'19)*. ACM, New York, NY, 2858–2864. https://doi.org/10.1145/3308558.3313576

[104] O. Jimenez-del-Toro, H. Muller, M. Krenn, K. Gruenberg, A. A. Taha, M. Winterstein, I. Eggel, et al. 2016. Cloud-based evaluation of anatomical structure segmentation and landmark detection algorithms: VISCERAL anatomy benchmarks. *IEEE Transactions on Medical Imaging* 35, 11 (2016), 2459–2475.

[105] Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2017. Billion-scale similarity search with GPUs. *arXiv preprint arXiv:1702.08734* (2017).

[106] Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2019. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data* 7, 3 (2019), 535–547.

[107] Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 1601–1611. https://doi.org/10.18653/v1/P17-1147

[108] Maryam Karimzadehgan and ChengXiang Zhai. 2010. Estimation of statistical translation models based on mutual information for ad hoc information retrieval. In *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'10)*. ACM, New York, NY, 323–330. https://doi.org/10.1145/1835449.1835505

[109] Maryam Karimzadehgan and ChengXiang Zhai. 2012. Axiomatic analysis of translation language model for information retrieval. In *Proceedings of the European Conference on Information Retrieval*. 268–280.

[110] Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-Tau Yih. 2020. Dense passage retrieval for open-domain question answering. *arXiv preprint arXiv:2004.04906* (2020).

[111] Tom Kenter and Maarten de Rijke. 2015. Short text similarity with word embeddings. In *Proceedings of the 24th ACM International Conference on Information and Knowledge Management (CIKM'15)*. ACM, New York, NY, 1411–1420. https://doi.org/10.1145/2806416.2806475

[112] Omar Khattab and Matei 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 (SIGIR'20)*. ACM, New York, NY, 39–48. https://doi.org/10.1145/3397271.3401075

[113] Jon M. Kleinberg. 2000. Navigation in a small world. *Nature* 406, 6798 (2000), 845–845.

[114] Oren Kurland and Lillian Lee. 2004. Corpus structure, language models, and ad hoc information retrieval. In *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. 194–201.

[115] Saar Kuzi, Mingyang Zhang, Cheng Li, Michael Bendersky, and Marc Najork. 2020. Leveraging semantic and lexical matching to improve the recall of document retrieval systems: A hybrid approach. *arXiv preprint arXiv:2010.01195* (2020).

[116] Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, et al. 2019. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics* 7 (March 2019), 452–466. https://doi.org/10.1162/tacl_a_00276

[117] Kui-Lam Kwok, Laszlo Grunfeld, H. L. Sun, Peter Deng, and N. Dinstl. 2004. TREC2004 robust track experiments using PIRCS. In *Proceedings of the 2004 Text REtrieval Conference (TREC'04)*.
[118] Victor Lavrenko. 2008. *A Generative Theory of Relevance*. Vol. 26. Springer Science & Business Media.
[119] Victor Lavrenko and W. Bruce Croft. 2001. Relevance based language models. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'01)*. ACM, New York, NY, 120–127. https://doi.org/10.1145/383952.383972
[120] Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *Proceedings of the International Conference on Machine Learning*. 1188–1196.
[121] Daniel D. Lee and H. Sebastian Seung. 2001. Algorithms for non-negative matrix factorization. In *Advances in Neural Information Processing Systems 13*, T. K. Leen, T. G. Dietterich, and V. Tresp (Eds.). MIT Press, Cambridge, MA, 556–562. http://papers.nips.cc/paper/1861-algorithms-for-non-negative-matrix-factorization.pdf.
[122] Jinhyuk Lee, Minjoon Seo, Hannaneh Hajishirzi, and Jaewoo Kang. 2020. Contextualized sparse representations for real-time open-domain question answering. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. 912–919. https://doi.org/10.18653/v1/2020.acl-main.85
[123] Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. 2019. Latent retrieval for weakly supervised open domain question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. 6086–6096. https://doi.org/10.18653/v1/P19-1612
[124] Michael E. Lesk. 1969. Word-word associations in document retrieval systems. *American Documentation* 20, 1 (1969), 27–38.
[125] Chao Li, Zhiyuan Liu, Mengmeng Wu, Yuchi Xu, Huan Zhao, Pipei Huang, Guoliang Kang, Qiwei Chen, Wei Li, and Dik Lun Lee. 2019. Multi-interest network with dynamic routing for recommendation at Tmall. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*. 2615–2623.
[126] Hang Li. 2011. Learning to rank for information retrieval and natural language processing. *Synthesis Lectures on Human Language Technologies* 4, 1 (2011), 1–113.
[127] Hang Li and Jun Xu. 2014. Semantic matching in search. *Foundations and Trends in Information Retrieval* 7, 5 (2014), 343–469.
[128] Rui Li, Yunjiang Jiang, Wenyun Yang, Guoyu Tang, Songlin Wang, Chaoyi Ma, Wei He, Xi Xiong, Yun Xiao, and Eric Yihong Zhao. 2019. From semantic retrieval to pairwise ranking: Applying deep learning in e-commerce search. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'19)*. ACM, New York, NY, 1383–1384. https://doi.org/10.1145/3331184.3331434
[129] Wen Li, Ying Zhang, Yifang Sun, Wei Wang, Mingjie Li, Wenjie Zhang, and Xuemin Lin. 2019. Approximate nearest neighbor search on high dimensional data-experiments, analyses, and improvement. *IEEE Transactions on Knowledge and Data Engineering* 32, 8 (2019), 1475–1488.
[130] Davis Liang, Peng Xu, Siamak Shakeri, Cicero Nogueira dos Santos, Ramesh Nallapati, Zhiheng Huang, and Bing Xiang. 2020. Embedding-based zero-shot retrieval through query generation. *arXiv preprint arXiv:2009.10270* (2020).
[131] Jimmy Lin, Rodrigo Nogueira, and Andrew Yates. 2020. Pretrained transformers for text ranking: Bert and beyond. *arXiv preprint arXiv:2010.06467* (2020).
[132] Sheng-Chieh Lin, Jheng-Hong Yang, and Jimmy Lin. 2020. Distilling dense representations for ranking using tightly-coupled teachers. *arXiv preprint arXiv:2010.11386* (2020).
[133] Shichen Liu, Fei Xiao, Wenwu Ou, and Luo Si. 2017. Cascade ranking for operational e-commerce search. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'17)*. ACM, New York, NY, 1557–1565. https://doi.org/10.1145/3097983.3098011
[134] Tie-Yan Liu. 2011. *Learning to Rank for Information Retrieval*. Springer Science & Business Media.
[135] Xiaoyong Liu and W. Bruce Croft. 2004. Cluster-based retrieval using language models. In *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. 186–193.
[136] Xiaojie Liu, Jian-Yun Nie, and Alessandro Sordoni. 2016. Constraining word embeddings by prior knowledge—Application to medical information retrieval. In *Proceedings of the Asia Information Retrieval Symposium*. 155–167.
[137] Yi Luan, Jacob Eisenstein, Kristina Toutanova, and Michael Collins. 2020. Sparse, dense, and attentional representations for text retrieval. *arXiv preprint arXiv:2005.00181* (2020).
[138] Yuanhua Lv and ChengXiang Zhai. 2009. A comparative study of methods for estimating query language models with pseudo feedback. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management*. 1895–1898.
[139] Xinyu Ma, Jiafeng Guo, Ruqing Zhang, Yixing Fan, Xiang Ji, and Xueqi Cheng. 2020. PROP: Pre-training with representative words prediction for ad-hoc retrieval. *arXiv preprint arXiv:2010.10137* (2020).
[140] Xinyu Ma, Jiafeng Guo, Ruqing Zhang, Yixing Fan, Yingyan Li, and Xueqi Cheng. 2021. B-PROP: Bootstrapped pre-training with representative words prediction for ad-hoc retrieval. *arXiv preprint arXiv:2104.09791* (2021).
[141] Sean MacAvaney, Franco Maria Nardini, Raffaele Perego, Nicola Tonellotto, Nazli Goharian, and Ophir Frieder. 2020. Efficient document re-ranking for transformers by precomputing term representations. In *Proceedings of the 43rd*

International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'20). ACM, New York, NY, 49–58. https://doi.org/10.1145/3397271.3401093

[142] Sean MacAvaney, Franco Maria Nardini, Raffaele Perego, Nicola Tonellotto, Nazli Goharian, and Ophir 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 (SIGIR'20)*. ACM, New York, NY, 1573–1576. https://doi.org/10.1145/3397271.3401262

[143] Joel Mackenzie, Zhuyun Dai, Luke Gallagher, and Jamie Callan. 2020. Efficiency implications of term weighting for passage retrieval. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'20)*. ACM, New York, NY, 1821–1824. https://doi.org/10.1145/3397271.3401263

[144] Yu A. Malkov and Dmitry A. Yashunin. 2018. Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 42, 4 (2018), 824–836.

[145] Antonio Mallia, Omar Khattab, Nicola Tonellotto, and Torsten Suel. 2021. Learning passage impacts for inverted indexes. *arXiv preprint arXiv:2104.12016* (2021).

[146] Yuning Mao, Pengcheng He, Xiaodong Liu, Yelong Shen, Jianfeng Gao, Jiawei Han, and Weizhu Chen. 2020. Generation-augmented retrieval for open-domain question answering. *arXiv preprint arXiv:2009.08553* (2020).

[147] Irina Matveeva, Chris Burges, Timo Burkard, Andy Laucius, and Leon Wong. 2006. High accuracy retrieval with multiple nested ranker. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'06)*. ACM, New York, NY, 437–444. https://doi.org/10.1145/1148170.1148246

[148] Donald Metzler and W. Bruce Croft. 2005. A Markov random field model for term dependencies. In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'05)*. ACM, New York, NY, 472–479. https://doi.org/10.1145/1076034.1076115

[149] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S. Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems 26*, C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger (Eds.). Curran Associates, 3111–3119. http://papers.nips.cc/paper/5021-distributed-representations-of-words-and-phrases-and-their-compositionality.pdf.

[150] David R. H. Miller, Tim Leek, and Richard M. Schwartz. 1999. A hidden Markov model information retrieval system. In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'99)*. ACM, New York, NY, 214–221. https://doi.org/10.1145/312624.312680

[151] Bhaskar Mitra and Nick Craswell. 2017. Neural models for information retrieval. *arXiv preprint arXiv:1705.01509* (2017).

[152] Bhaskar Mitra and Nick Craswell. 2018. An introduction to neural information retrieval. *Foundations and Trends in Information Retrieval* 13, 1 (2018), 1–126.

[153] Bhaskar Mitra, Fernando Diaz, and Nick 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 (WWW'17)*. 1291–1299. https://doi.org/10.1145/3038912.3052579

[154] Bhaskar Mitra, Sebastian Hofstatter, Hamed Zamani, and Nick Craswell. 2020. Conformer-kernel with query term independence for document retrieval. *arXiv preprint arXiv:2007.10434* (2020).

[155] Bhaskar Mitra, Eric Nalisnick, Nick Craswell, and Rich Caruana. 2016. A dual embedding space model for document ranking. *arXiv preprint arXiv:1602.01137* (2016).

[156] Bhaskar Mitra, Corby Rosset, David Hawking, Nick Craswell, Fernando Diaz, and Emine Yilmaz. 2019. Incorporating query term independence assumption for efficient retrieval and ranking using deep neural networks. *arXiv preprint arXiv:1907.03693* (2019).

[157] Marius Muja and David G. Lowe. 2014. Scalable nearest neighbor algorithms for high dimensional data. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 36, 11 (2014), 2227–2240.

[158] Eric Nalisnick, Bhaskar Mitra, Nick Craswell, and Rich Caruana. 2016. Improving document ranking with dual word embeddings. In *Proceedings of the 25th International Conference Companion on World Wide Web (WWW'16 Companion)*. 83–84. https://doi.org/10.1145/2872518.2889361

[159] Gia-Hung Nguyen, Lynda Tamine, Laure Soulier, and Nathalie Souf. 2017. Learning concept-driven document embeddings for medical information search. In *Proceedings of the Conference on Artificial Intelligence in Medicine in Europe*. 160–170.

[160] Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. MS MARCO: A human generated machine reading comprehension dataset. In *Proceedings of the Workshop on Cognitive Computation: Integrating Neural and Symbolic Approaches 2016 (CEUR Workshop Proceedings, Vol. 1773)*, Tarek Richard Besold, Antoine Bordes, Artur S. d'Avila Garcez, and Greg Wayne (Eds.). CEUR-WS.org, 1–10. http://ceur-ws.org/Vol-1773/CoCoNIPS_2016_paper9.pdf.

[161] Rodrigo Nogueira and Kyunghyun Cho. 2019. Passage re-ranking with BERT. *arXiv preprint arXiv:1901.04085* (2019).

[162] Rodrigo Nogueira and Jimmy Lin. 2019. *From doc2query to docTTTTTquery*. Technical Report. MS MARCO Passage Retrieval Task Publication.
[163] Rodrigo Nogueira, Wei Yang, Kyunghyun Cho, and Jimmy Lin. 2019. Multi-stage document ranking with BERT. *arXiv preprint arXiv:1910.14424* (2019).
[164] Rodrigo Nogueira, Wei Yang, Jimmy Lin, and Kyunghyun Cho. 2019. Document expansion by query prediction. *arXiv preprint arXiv:1904.08375* (2019).
[165] Kezban Dilek Onal, Ye Zhang, Ismail Sengor Altingovde, Md. Mustafizur Rahman, Pinar Karagoz, Alex Braylan, et al. 2018. Neural information retrieval: At the end of the early years. *Information Retrieval Journal* 21, 2–3 (2018), 111–182.
[166] Dae Hoon Park and Yi Chang. 2019. Adversarial sampling and training for semi-supervised information retrieval. In *Proceedings of the World Wide Web Conference (WWW'19)*. ACM, New York, NY, 1443–1453. https://doi.org/10.1145/3308558.3313416
[167] Jan Pedersen. 2010. Query understanding at bing. Invited Talk, SIGIR (2010).
[168] Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP'14)*. 1532–1543. https://doi.org/10.3115/v1/D14-1162
[169] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365* (2018).
[170] Jay M. Ponte and W. Bruce Croft. 1998. A language modeling approach to information retrieval. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'98)*. ACM, New York, NY, 275–281. https://doi.org/10.1145/290941.291008
[171] Yonggang Qiu and Hans-Peter Frei. 1993. Concept based query expansion. In *Proceedings of the 16th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'93)*. ACM, New York, NY, 160–169. https://doi.org/10.1145/160688.160713
[172] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. Preprint.
[173] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research* 21, 140 (2020), 1–67. http://jmlr.org/papers/v21/20-074.html.
[174] Ankit Singh Rawat, Jiecao Chen, Felix Xinnan X. Yu, Ananda Theertha Suresh, and Sanjiv Kumar. 2019. Sampled softmax with random Fourier features. In *Advances in Neural Information Processing Systems*, Vol. 32. Curran Associates, 13857–13867. https://proceedings.neurips.cc/paper/2019/file/e43739bba7cdb577e9e3e4e42447f5a5-Paper.pdf.
[175] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian personalized ranking from implicit feedback. In *Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence (UAI'09)*. 452–461.
[176] Stefan Riezler and Yi Liu. 2010. Query rewriting using monolingual statistical machine translation. *Computational Linguistics* 36, 3 (Sept. 2010), 569–582. https://doi.org/10.1162/coli_a_00010
[177] Stephen Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: BM25 and beyond. *Foundations and Trends in Information Retrieval* 3, 4 (April 2009), 333–389. https://doi.org/10.1561/1500000019
[178] Stephen E. Robertson and Karen Sparck Jones. 1988. *Relevance Weighting of Search Terms*. Taylor Graham Publishing, GBR, 143–160.
[179] Joseph Rocchio. 1971. Relevance feedback in information retrieval. *Smart Retrieval System—Experiments in Automatic Document Processing* 1971 (1971), 313–323.
[180] Dwaipayan Roy, Debasis Ganguly, Mandar Mitra, and Gareth J. F. Jones. 2016. Representing documents and queries as sets of word embedded vectors for information retrieval. In *Proceedings of Neu-IR: The SIGIR 2016 Workshop on Neural Information Retrieval*.
[181] Dwaipayan Roy, Debjyoti Paul, Mandar Mitra, and Utpal Garain. 2016. Using word embeddings for automatic query expansion. In *Proceedings of Neu-IR: The SIGIR 2016 Workshop on Neural Information Retrieval*.
[182] Ruslan Salakhutdinov and Geoffrey Hinton. 2009. Semantic hashing. *International Journal of Approximate Reasoning* 50, 7 (2009), 969–978.
[183] Gerard Salton. 1991. Developments in automatic text retrieval. *Science* 253, 5023 (1991), 974–980.
[184] Gerard Salton and Christopher Buckley. 1988. Term-weighting approaches in automatic text retrieval. *Information Processing and Management* 24, 5 (Aug. 1988), 513–523. https://doi.org/10.1016/0306-4573(88)90021-0
[185] G. Salton, A. Wong, and C. S. Yang. 1975. A vector space model for automatic indexing. *Communications of the ACM* 18, 11 (Nov. 1975), 613–620. https://doi.org/10.1145/361219.361220
[186] Minjoon Seo, Tom Kwiatkowski, Ankur Parikh, Ali Farhadi, and Hannaneh Hajishirzi. 2018. Phrase-indexed question answering: A new challenge for scalable document comprehension. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. 559–564. https://doi.org/10.18653/v1/D18-1052

[187] Minjoon Seo, Jinhyuk Lee, Tom Kwiatkowski, Ankur Parikh, Ali Farhadi, and Hannaneh Hajishirzi. 2019. Real-time open-domain question answering with dense-sparse phrase index. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. 4430–4441. https://doi.org/10.18653/v1/P19-1436

[188] Gregory Shakhnarovich, Trevor Darrell, and Piotr Indyk. 2006. *Nearest-Neighbor Methods in Learning and Vision: Theory and Practice (Neural Information Processing)*. MIT Press, Cambridge, MA.

[189] Yelong Shen, Xiaodong He, Jianfeng Gao, Li Deng, and Grégoire Mesnil. 2014. Learning semantic representations using convolutional neural networks for web search. In *Proceedings of the 23rd International Conference on World Wide Web (WWW'14 Companion)*. ACM, New York, NY, 373–374. https://doi.org/10.1145/2567948.2577348

[190] Garrick Sherman and Miles Efron. 2017. Document expansion using external collections. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1045–1048.

[191] R. F. Simmons. 1965. Answering English questions by computer: A survey. *Communications of the ACM* 8, 1 (Jan. 1965), 53–70. https://doi.org/10.1145/363707.363732

[192] Amit Singhal and Fernando Pereira. 1999. Document expansion for speech retrieval. In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. 34–41.

[193] Ivan Srba and Maria Bielikova. 2016. A comprehensive survey and classification of approaches for community question answering. *ACM Transactions on the Web* 10, 3 (Aug. 2016), Article 18, 63 pages. https://doi.org/10.1145/2934687

[194] Yifan Sun, Changmao Cheng, Yuhan Zhang, Chi Zhang, Liang Zheng, Zhongdao Wang, and Yichen Wei. 2020. Circle loss: A unified perspective of pair similarity optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR'20)*. 6398–6407.

[195] Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. 2019. RotatE: Knowledge graph embedding by relational rotation in complex space. In *Proceedings of the International Conference on Learning Representations*. https://openreview.net/forum?id=HkgEQnRqYQ.

[196] Amir Vakili Tahami, Kamyar Ghajar, and Azadeh Shakery. 2020. Distilling knowledge for fast retrieval-based chatbots. *arXiv preprint arXiv:2004.11045* (2020).

[197] Lynda Tamine, Laure Soulier, Gia-Hung Nguyen, and Nathalie Souf. 2019. Offline versus online representation learning of documents using external knowledge. *ACM Transactions on Information Systems* 37, 4 (2019), 1–34.

[198] Ming Tan, Cicero dos Santos, Bing Xiang, and Bowen Zhou. 2015. LSTM-based deep learning models for non-factoid answer selection. *arXiv preprint arXiv:1511.04108* (2015).

[199] Hongyin Tang, Xingwu Sun, Beihong Jin, Jingang Wang, Fuzheng Zhang, and Wei Wu. 2021. Improving document representations by generating pseudo query embeddings for dense retrieval. *arXiv preprint arXiv:2105.03599* (2021).

[200] Tao Tao, Xuanhui Wang, Qiaozhu Mei, and ChengXiang Zhai. 2006. Language model information retrieval with document expansion. In *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*. 407–414.

[201] Aäron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *CoRR* abs/1807.03748 (2018). arXiv:1807.03748 http://arxiv.org/abs/1807.03748.

[202] Cornelis Joost Van Rijsbergen. 1977. A theoretical basis for the use of co-occurrence data in information retrieval. *Journal of Documentation* 33, 2 (1977), 106–199.

[203] Ellen Voorhees, Tasmeer Alam, Steven Bedrick, Dina Demner-Fushman, William R. Hersh, Kyle Lo, Kirk Roberts, Ian Soboroff, and Lucy Lu Wang. 2021. TREC-COVID: Constructing a pandemic information retrieval test collection. *ACM SIGIR Forum* 54 (2021), 1–12.

[204] Ellen M. Voorhees. 1994. Query expansion using lexical-semantic relations. In *Proceedings of the 1994 SIGIR Conference*. 61–69.

[205] Ellen M. Voorhees. 2005. Overview of the TREC 2005 robust retrieval track. In *Proceedings of the 2005 Text REtrieval Conference (TREC'05)*.

[206] Ellen M. Voorhees and Dawn M. Tice. 2000. Building a question answering test collection. In *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'00)*. ACM, New York, NY, 200–207. https://doi.org/10.1145/345508.345577

[207] Ivan Vulić and Marie-Francine Moens. 2015. Monolingual and cross-lingual information retrieval models based on (bilingual) word embeddings. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'15)*. ACM, New York, NY, 363–372. https://doi.org/10.1145/2766462.2767752

[208] Shengxian Wan, Yanyan Lan, Jun Xu, Jiafeng Guo, Liang Pang, and Xueqi Cheng. 2016. Match-SRNN: Modeling the recursive matching structure with spatial RNN. *arXiv preprint arXiv:1604.04378* (2016).

[209] Feng Wang, Weiyang Liu, Haijun Liu, and Jian Cheng. 2018. Additive margin softmax for face verification. *IEEE Signal Processing Letters* 25, 7 (2018), 926–930. https://doi.org/10.1109/LSP.2018.2822810

[210] Feng Wang, Xiang Xiang, Jian Cheng, and Alan Loddon Yuille. 2017. NormFace: L_2 hypersphere embedding for face verification. In *Proceedings of the 25th ACM International Conference on Multimedia (MM'17)*. ACM, New York, NY, 1041–1049. https://doi.org/10.1145/3123266.3123359

[211] Quan Wang, Jun Xu, Hang Li, and Nick Craswell. 2011. Regularized latent semantic indexing. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'11)*. ACM, New York, NY, 685–694. https://doi.org/10.1145/2009916.2010008

[212] Zizhen Wang, Yixing Fan, Jiafeng Guo, Liu Yang, Ruqing Zhang, Yanyan Lan, Xueqi Cheng, Hui Jiang, and Xiaozhao Wang. 2020. Match^2: A matching over matching model for similar question identification. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 559–568.

[213] Xing Wei and W. Bruce Croft. 2006. LDA-based document models for ad-hoc retrieval. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'06)*. ACM, New York, NY, 178–185. https://doi.org/10.1145/1148170.1148204

[214] Ian H. Witten, Alistair Moffat, and Timothy C. Bell. 1999. *Managing Gigabytes: Compressing and Indexing Documents and Images*. Morgan Kaufmann.

[215] Chenyan Xiong, Zhuyun Dai, Jamie Callan, Zhiyuan Liu, and Russell Power. 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 (SIGIR'17)*. ACM, New York, NY, 55–64. https://doi.org/10.1145/3077136.3080809

[216] Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul Bennett, Junaid Ahmed, and Arnold Overwijk. 2020. Approximate nearest neighbor negative contrastive learning for dense text retrieval. *CoRR* abs/2007.00808 (2020). arXiv:2007.00808 https://arxiv.org/abs/2007.00808.

[217] Jinxi Xu and W. Bruce Croft. 2017. Query expansion using local and global document analysis. *ACM SIGIR Forum* 51 (2017), 168–175.

[218] Jun Xu, Hang Li, and Chaoliang Zhong. 2010. Relevance ranking using kernels. In *Proceedings of the Asia Information Retrieval Symposium*. 1–12.

[219] Ming Yan, Chenliang Li, Bin Bi, Wei Wang, and Songfang Huang. 2021. A unified pretraining framework for passage ranking and expansion. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 4555–4563.

[220] Ji Yang, Xinyang Yi, Derek Zhiyuan Cheng, Lichan Hong, Yang Li, Simon Xiaoming Wang, Taibai Xu, and Ed H. Chi. 2020. Mixed negative sampling for learning two-tower neural networks in recommendations. In *Companion Proceedings of the Web Conference 2020 (WWW'20)*. ACM, New York, NY, 441–447. https://doi.org/10.1145/3366424.3386195

[221] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R. Salakhutdinov, and Quoc V. Le. 2019. XLNet: Generalized autoregressive pretraining for language understanding. In *Advances in Neural Information Processing Systems 32*. Curran Associates, 5753–5763. http://papers.nips.cc/paper/8812-xlnet-generalized-autoregressive-pretraining-for-language-understanding.pdf.

[222] Zhen Yang, Ming Ding, Chang Zhou, Hongxia Yang, Jingren Zhou, and Jie Tang. 2020. Understanding negative sampling in graph representation learning. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, New York, NY, 1666–1676. https://doi.org/10.1145/3394486.3403218

[223] Liang Yao, Chengsheng Mao, and Yuan Luo. 2019. Graph convolutional networks for text classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 7370–7377.

[224] Xing Yi and James Allan. 2009. A comparative study of utilizing topic models for information retrieval. In *Advances in Information Retrieval*, Mohand Boughanem, Catherine Berrut, Josiane Mothe, and Chantal Soule-Dupuy (Eds.). Springer, Berlin, Germany, 29–41.

[225] Xinyang Yi, Ji Yang, Lichan Hong, Derek Zhiyuan Cheng, Lukasz Heldt, Aditee Kumthekar, Zhe Zhao, Li Wei, and Ed Chi. 2019. Sampling-bias-corrected neural modeling for large corpus item recommendations. In *Proceedings of the 13th ACM Conference on Recommender Systems (RecSys'19)*. ACM, New York, NY, 269–277. https://doi.org/10.1145/3298689.3346996

[226] Rex Ying, Ruining He, Kaifeng Chen, Pong Eksombatchai, William L. Hamilton, and Jure Leskovec. 2018. Graph convolutional neural networks for web-scale recommender systems. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'18)*. ACM, New York, NY, 974–983. https://doi.org/10.1145/3219819.3219890

[227] Tan Yu, Junsong Yuan, Chen Fang, and Hailin Jin. 2018. Product quantization network for fast image retrieval. In *Proceedings of the European Conference on Computer Vision (ECCV'18)*. 186–201.

[228] Hamed Zamani, Mostafa Dehghani, W. Bruce Croft, Erik Learned-Miller, and Jaap Kamps. 2018. From neural re-ranking to neural ranking: Learning a sparse representation for inverted indexing. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management (CIKM'18)*. ACM, New York, NY, 497–506. https://doi.org/10.1145/3269206.3271800

[229] ChengXiang Zhai. 2008. Statistical language models for information retrieval a critical review. *Foundations and Trends in Information Retrieval* 2, 3 (March 2008), 137–213. https://doi.org/10.1561/1500000008

[230] Chengxiang Zhai and John Lafferty. 2001. Model-based feedback in the language modeling approach to information retrieval. In *Proceedings of the 10th International Conference on Information and Knowledge Management*. 403–410.

[231] Jingtao Zhan, Jiaxin Mao, Yiqun Liu, Jiafeng Guo, Min Zhang, and Shaoping Ma. 2021. Optimizing dense retrieval model training with hard negatives. *arXiv preprint arXiv:2104.08051* (2021).

[232] Jingtao Zhan, Jiaxin Mao, Yiqun Liu, Min Zhang, and Shaoping Ma. 2020. RepBERT: Contextualized text embeddings for first-stage retrieval. *CoRR* abs/2006.15498 (2020). https://arxiv.org/abs/2006.15498.

[233] Han Zhang, Hongwei Shen, Yiming Qiu, Yunjiang Jiang, Songlin Wang, Sulong Xu, Yun Xiao, Bo Long, and Wen-Yun Yang. 2021. Joint learning of deep retrieval model and product quantization based embedding index. *arXiv preprint arXiv:2105.03933* (2021).

[234] Han Zhang, Songlin Wang, Kang Zhang, Zhiling Tang, Yunjiang Jiang, Yun Xiao, Weipeng Yan, and Wen-Yun Yang. 2020. Towards personalized and semantic retrieval: An end-to-end solution for e-commerce search via embedding learning. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'20)*. ACM, New York, NY, 2407–2416. https://doi.org/10.1145/3397271.3401446

[235] Minjia Zhang and Yuxiong He. 2019. GRIP: Multi-store capacity-optimized high-performance nearest neighbor search for vector search engine. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management (CIKM'19)*. ACM, New York, NY, 1673–1682. https://doi.org/10.1145/3357384.3357938

[236] Weinan Zhang, Tianqi Chen, Jun Wang, and Yong Yu. 2013. Optimizing top-n collaborative filtering via dynamic negative item sampling. In *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'13)*. ACM, New York, NY, 785–788. https://doi.org/10.1145/2484028.2484126

[237] Yuyu Zhang, Ping Nie, Xiubo Geng, Arun Ramamurthy, Le Song, and Daxin Jiang. 2020. DC-BERT: Decoupling question and document for efficient contextual encoding. *arXiv preprint arXiv:2002.12591* (2020).

[238] Zheng Zhang and Pierre Zweigenbaum. 2018. GNEG: Graph-based negative sampling for Word2Vec. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. 566–571. https://doi.org/10.18653/v1/P18-2090

[239] Le Zhao and Jamie Callan. 2010. Term necessity prediction. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*. 259–268.

[240] Guoqing Zheng and Jamie Callan. 2015. Learning to reweight terms with distributed representations. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'15)*. ACM, New York, NY, 575–584. https://doi.org/10.1145/2766462.2767700

[241] Justin Zobel and Alistair Moffat. 2006. Inverted files for text search engines. *ACM Computing Surveys* 38, 2 (2006), 6–es.

[242] Guido Zuccon, Bevan Koopman, Peter Bruza, and Leif Azzopardi. 2015. Integrating and evaluating neural word embeddings in information retrieval. In *Proceedings of the 20th Australasian Document Computing Symposium (ADCS'15)*. ACM, New York, NY, Article 12, 8 pages. https://doi.org/10.1145/2838931.2838936

Received March 2021; revised August 2021; accepted September 2021