Pre-training Methods in Information Retrieval

Suggested Citation: Yixing Fan*, Xiaohui Xie*, Yinqiong Cai, Jia Chen, Xinyu Ma, Xiangsheng Li, Ruqing Zhang, Jiafeng Guo* and Yiqun Liu* (2021), "Pre-training Methods in Information Retrieval", : Vol. xx, No. xx, pp 1–18. DOI: 10.1561/XXXXXXXXX.

Yixing Fan
ICT, CAS, China
fanyixing@ict.ac.cn

Xiaohui Xie
Tsinghua University
xiexiaohui@mail.tsinghua.edu.cn

Yinqiong Cai
ICT, CAS, China
caiyinqiong18s@ict.ac.cn

Jia Chen
Tsinghua University
chenjia0831@gmail.com

Xinyu Ma
ICT, CAS, China
maxinyu17g@ict.ac.cn

Xiangsheng Li
Tsinghua University
lixsh6@gmail.com

Ruqing Zhang
ICT, CAS, China
zhangruqing@ict.ac.cn

Jiafeng Guo
ICT, CAS, China
guojiafeng@ict.ac.cn

Yiqun Liu
Tsinghua University
yiqunliu@tsinghua.edu.cn

now
the essence of knowledge
Boston — Delft

Contents

Pre-training Methods in Information Retrieval

Yixing Fan*[1], Xiaohui Xie*[2], Yinqiong Cai[1], Jia Chen[2], Xinyu Ma[1], Xiangsheng Li[2], Ruqing Zhang[1], Jiafeng Guo⋆[1] and Yiqun Liu⋆[2]

[1] *ICT, CAS, China; fanyixing@ict.ac.cn*
[2] *Tsinghua University; xiexiaohui@mail.tsinghua.edu.cn*
[1] *ICT, CAS, China; caiyinqiong18s@ict.ac.cn*
[2] *Tsinghua University; chenjia0831@gmail.com*
[1] *ICT, CAS, China; maxinyu17g@ict.ac.cn*
[2] *Tsinghua University; lixsh6@gmail.com*
[1] *ICT, CAS, China; zhangruqing@ict.ac.cn*
[1] *ICT, CAS, China; guojiafeng@ict.ac.cn*
[2] *Tsinghua University; yiqunliu@tsinghua.edu.cn*

ABSTRACT

The core of information retrieval (IR) is to identify relevant information from large-scale resources and return it as a ranked list to respond to user's information need. Recently, the resurgence of deep learning has greatly advanced this field and leads to a hot topic named NeuIR (i.e., neural information retrieval), especially the paradigm of pre-training methods (PTMs). Owing to sophisticated pre-training objectives and huge model size, pre-trained models can learn universal language representations from massive textual data, which are beneficial to the ranking task of IR. Since there have been a large number of works dedicating to the

* Yixing Fan and Xiaohui Xie contributed equally.
⋆ Corresponding authors.

Yixing Fan*, Xiaohui Xie*, Yinqiong Cai, Jia Chen, Xinyu Ma, Xiangsheng Li, Ruqing Zhang, Jiafeng Guo⋆ and Yiqun Liu⋆ (2021), "Pre-training Methods in Information Retrieval", : Vol. xx, No. xx, pp 1–18. DOI: 10.1561/XXXXXXXXX.

application of PTMs in IR, we believe it is the right time to summarize the current status, learn from existing methods, and gain some insights for future development. In this survey, we present an overview of PTMs applied in different components of IR system, including the retrieval component, the re-ranking component, and other components. In addition, we also introduce PTMs specifically designed for IR, and summarize available datasets as well as benchmark leaderboards. Moreover, we discuss some open challenges and envision some promising directions, with the hope of inspiring more works on these topics for future research.

1

Introduction

Information retrieval (IR) is a fundamental task in many real-world applications, such as digital libraries, Web search, question answering systems, and so on. The core of IR is to identify information resources relevant to user's information need (e.g., query or question) from a large collection. Since there might be a variety of relevant resources, the returned result is often a ranked list of documents according to their relevance degree against the information need. Such ranking property of IR makes it different from other tasks, and researchers have devoted substantial efforts to develop a variety of ranking models in IR.

Over the past decades, many different ranking models have been proposed and studied, including vector space models (Salton *et al.*, 1975), probabilistic models (Robertson and Jones, 1976b), and learning to rank (LTR) models (Li, 2011). These methods have been successfully applied in many different IR applications, such as Web search engines like Google, news recommend systems like Toutiao, community question answering platform like Quora, to name a few. More recently, a large variety of neural ranking models have been proposed, leading to a hot topic named NeuIR (i.e., neural information retrieval). Different from previous non-neural models that rely on hand-crafted features and statistical methods, neural ranking models can automatically learn low-dimensional continuous vectors (a.k.a., latent representations) from data as ranking features, thereby get rid of complex feature engineering. Despite the success of neural models in IR, a major performance bottleneck lies in the availability of large scale, high-quality and labeled datasets as deep neural models often have a large number of parameters to learn.

In recent years, Pre-training Methods (PTMs) have brought a storm and fueled a paradigm shift in Nature Language Processing (NLP) (Qiu *et al.*, 2020). The idea is to firstly pre-train models in large-scale corpus through self-supervised training objectives, and then fine-tune the pre-trained models for various downstream tasks to achieve state-of-the-art performances. As is demonstrated in recent works (Peters *et al.*, 2018b; Howard and Ruder, 2018), these pre-trained models are able to capture decent amount of lexical knowledge as well as factual knowledge, which are beneficial for downstream tasks and can void learning such knowledge from scratch. Moreover, with the increasing amount of computational power and the emergence of Transformer architecture (Vaswani *et al.*, 2017), we can further advance the parameter scale of pre-trained models from million-level to billion-level (e.g., BERT (Devlin *et al.*, 2018) and GPT-3 (Brown *et al.*, 2020b)) and even trillion-level (e.g., Switch-Transformers (Fedus *et al.*, 2021)). Both of these are desirable properties for ranking models in IR. On on hand, pre-trained models, which are pre-trained on huge textual corpus with self-supervised modeling objectives, are able to better understand intents behind queries and semantics of documents. On the other hand, large-scale pre-trained models with deeply stacked Transformers have sufficient modeling capacities to learn complicated relevance patterns between queries and documents. Due to these potential benefits and along with the expectation that similar success with PTMs could be achieved in IR, we have witnessed explosive growth of research interest in exploiting PTMs in IR (Croft *et al.*, 2009; Manning *et al.*, 2005). Note that in this survey, we focus on PTMs in textual retrieval, which is central to IR. Readers who are interested in PTMs in content-based image retrieval (Dubey, 2020) or multi-modal retrieval could refer to (Fei *et al.*, 2021).

Up to now, numerous studies have been devoted to the application of PTMs in IR. In academic, researchers have carried out a variety of innovation and initiative in the usage of PTMs in IR. For example, earlier attempts tried to transfer knowledges learned from pre-trained models directly to ranking models, and have achieved some notable results (Nogueira and Cho, 2019; Dai and Callan, 2019b). More recent works proposed to promote existing pre-trained models by either reform the model architecture (MacAvaney *et al.*, 2020; Khattab and Zaharia, 2020; Gao and Callan, 2021a) or consider novel pre-training objectives (Chang *et al.*, 2019; Ma *et al.*, 2021b; Ma *et al.*, 2021c), which better meet the requirements of IR. Meanwhile, in industry, Google's October 2019 blog post and Bing's November 2019 blog post both showed that pre-trained ranking models (e.g., BERT-based models) can better understand the query intent and deliver a more useful result in practical search systems. Besides, looking at the ranking leaderboard today, we can see that most top-ranked methods are built on PTMs, just by looking at the names of these submissions. Considering the increasing number of studies on PTMs in IR, we believe that it is the right time to survey current status, learn from existing methods, and gain some insights for future development.

In this survey, we aim to provide a systematic and comprehensive review of works about PTMs in IR. It covers PTMs published in major conferences (e.g., SIGIR, TheWebConf, CIKM, WSDM, ICLR, AAAI, ACL, EMNLP, and ECIR) and journals (e.g., TOIS, TKDE, TIST, IP&M, and TACL) in the fields of deep learning, natural language processing, and information retrieval from the year 2018 to 2021. There exists some previous works discussing related topics. For example, both Onal *et al.* (2018b) and Guo *et al.* (2020b) reviewed the landscape of neural IR research, paying specific attention to the application of neural methods to different IR tasks, but did not cover every aspect of PTMs in IR. Yates *et al.* (2021) provided an early survey of pre-trained models for IR, which focused mainly on the application of BERT in text ranking. Cai *et al.* (2021b) reviewed semantic models for the first-stage retrieval under a unified framework, including early semantic retrieval models, neural retrieval models, and retrieval models based on PTMs. Different

---

https://www.blog.google/products/search/search-language-understanding-bert/
https://azure.microsoft.com/en-us/blog/bing-delivers-its-largest-improvement-in-search-expe
https://microsoft.github.io/msmarco/#docranking

from these works, we make a systematic and comprehensive overview of PTMs applied in different components of IR system, including the first-stage retrieval component, the re-ranking component, and other components. We also describe PTMs specifically designed for IR tasks, as well as resources for pre-training or fine-tuning ranking models. In addition to the model discussion, we also introduce some open challenges and suggest potentially promising directions for future works.

The structure of this survey is organized as follows. We will firstly provide a systematic overview of IR in section 2. Following this, we then review works about PTMs applied in the retrieval component, the re-ranking component, and other components in sections 3 to 5, respectively. In section 6, we present works in designing novel PTMs tailored for IR. We also summarize available large-scale datasets as well as popular benchmark leaderboards in section 7. Finally, we conclude this paper in section 8 and raise some promising directions for future research.

2

Background

In this section, we describe basic concepts and definitions of IR in a hierarchical manner and briefly review PTMs in IR. This background overview can help readers gain basic ideas of IR and lead to a better understanding on how can PTMs be beneficial for IR.

As is shown in Figure 2.1, we illustrate IR by decomposing the search process with a hierarchical view, from the core problem to the framework, to the system. Specifically, we use capital letters Q, D, F to denote a set of queries, documents and retrieval functions, and lower-case letters q, d, f denote a specific instance respectively. R refers to relevance estimation models and r denotes returned search results against an issued query.

2.1 The Core Problem View of IR

The basic objective of the IR system is to provide relevant information to users in response to their information need. Thus, the most fundamental problem is to estimate the degree of relevance between a query q and a document d. In practice, search begins with the emergence of a user intent which is the main goal a user has when issuing a query into a search engine. To some extent, the query can be regarded as the

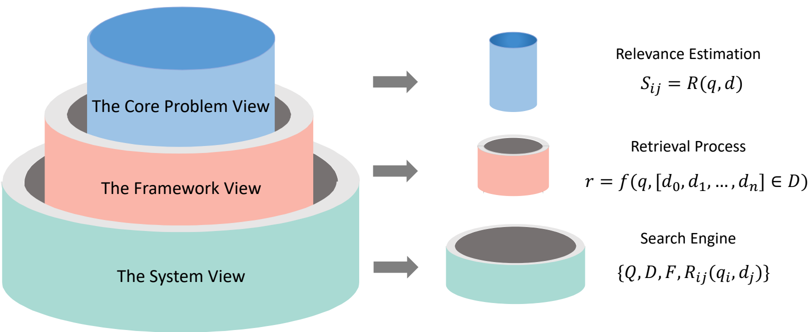

Figure 2.1: A Hierarchical View of IR

representative of the search intent. Then the mission of the search engine is to return the most "relevant" results related to the given query and display these results as a ranked list to the user. Thus, the better performance of the search engine in terms of estimating the relevance level between q and d leads to better user satisfaction. To evaluate the relevance score of a pair of q and d, existing works construct models to consider the correlation between the content of q and d on the basis of different strategies. There are three typical groups of these models:

- **Classical retrieval models**: The key idea of these models is to utilize exact matching signals to design a relevance scoring function. Specifically, these models consider easily computed statistics (e.g., term frequency, document length, and inverse document frequency) of normalized terms matched exactly between q and d. And the relevance score is derived from the sum of contributions from each query term that appears in the document. Among these models, BM25 (Robertson *et al.*, 1995) is shown to be effective and still be regarded as a strong baseline of many retrieval models nowadays. Besides BM25 and its variants, there are other representative retrieval functions, such as PIV (Singhal *et al.*, 2017) derived from vector space model, DIR (Zhai and Lafferty, 2004) derived using the language modeling approach, PL2 (Amati and Van Rijsbergen, 2002) based on the divergence from randomness framework, etc. However, these models may encounter the "vocabulary mismatch problem" due to "hard" and exact matching requirements.

- **Learning to Rank (LTR) Models**: The key idea of these models is to apply supervised machine learning technique to solve ranking problems using hand-crafted, manually-engineered features. Effective features include query-based features (e.g., query type and query length), document-based features (e.g., PageRank, document length, number of in-links and number of clicks) and query-document matching features (e.g., number of occurrences, BM25, N-gram BM25 and edit distance). According to the number of documents considered in loss functions, LTR models can be grouped into three basic types: 1) Pointwise approaches which considers individual documents and regard the retrieval problem as classification or regression problem. Example models include PRank (Perceptron Ranking) (Crammer, Singer, *et al.*, 2001) and McRank (Li *et al.*, 2007). 2) Pairwise approaches which take pairs of documents into consideration. For example, RankNet (Burges *et al.*, 2005b) is a pairwise method which adopts Cross Entropy as loss function in learning and RankSVM (Herbrich *et al.*, 1999) which transforms ranking into pairwise classification and employ the SVM technique to perform the learning task. 3) Listwise approaches which consider the entire list of documents. For example, LambdaMart (Burges *et al.*, 2006) trains a ranking function by employing Gradient Descent to minimize a listwise loss function. Please refer to another survey (Li, 2014) on LTR models for IR for more details.

- **Neural Retrieval Models**: The key idea of these models is to utilize neural networks to abstract relevance signals for relevance estimation. These models use the embedding of q and d as the input and are usually trained in an end-to-end manner with relevance labels. Compared to non-neural models, these models can be trained without handcrafted features. Without loss of generalities, these models can be grouped into representation-focused models, interaction-focused models, and mixed models. 1) Representation-focused models focus on independently learning dense vector representations of queries and documents. Then metrics such as cosine similarities and inner products are used to calculate the "distance" between queries and documents to access the relevance score. Example representation-focused models include DSSM (Huang *et al.*, 2013) and CDSSM (Shen *et al.*, 2014), etc. 2) Interaction-focused models capture "interactions" between queries and documents. These models utilize a similarity matrix A in which each entry A_{ij} refers to the similarity between embeddings of the i-th query term and the embedding of the j-th document term. After constructing the similarity matrix, interaction-based models apply different approaches to extract features that are adopted to produce the query-document relevance score. Example interaction-focused models include DRMM (Guo *et al.*, 2016b) and convKNRM (Xiong *et al.*, 2017b), etc. 3) Mixed models combine the design of the representation-focused component and the interaction-focused component, Duet (Mitra *et al.*, 2017) and CEDR (MacAvaney *et al.*, 2019b) for example. More detailed information can refer to these earlier surveys (Onal *et al.*, 2018a; Guo *et al.*, 2020a) on NeuIR models for IR

Due to the great success achieved by BERT (Devlin *et al.*, 2018) and its successors in NLP scenarios. Most PTMs in IR are also built on the basis of the Transformer framework to estimate the relevance level between the query and the document. These PTMs also have different high-level architectures, such as representation-focused (e.g., DPR (Karpukhin *et al.*, 2020), ColBERT (Khattab and Zaharia, 2020) and ME-BERT (Luan *et al.*, 2020)) and interaction-focus (e.g., MonoBERT (Nogueira and Cho, 2019), CEDR (MacAvaney *et al.*, 2019b) and duoBERT (Pradeep *et al.*, 2021)). For example, DPR (representation-focused) learns dense embeddings for the document with a BERT-based encoder, and queries are encoded with another independent BERT-based encoder. The outputs of two encoders are then fed into a "similarity" function to obtain the relevance score. MonoBERT (interaction-focused) takes the concatenating query and document as the input and feeds the [CLS] vector output by BERT to a feed-forward network to obtain the relevance score of the given query and document. More details can be found in the following sections.

2.2 The Framework View of IR

Given a document collection D, the aim of IR is to provide a search result list r where results are ordered in terms of their relevance levels given a query q. Since the document collection is massive, besides considering effectiveness, a practical IR system needs to give consideration to efficiency as well. In that regard, in a conventional retrieval architecture, several stages with different focuses on effectiveness and efficiency are built. We depict a retrieval architecture (f in Figure 2.1) in Figure 2.2. As shown in Figure 2.2, an initial retriever is involved to recall relevant results from a large document collection. In terms of relevance scores given by the retriever, these initial results are ranked to form an initial result list. Then this initial result list is passed through n re-rankers to generate the final ranked list which is provided to search users. Each re-ranker receives a ranked list from the previous stage and in turn provides a re-ranked list that contains the same number of or fewer results. Although both aiming at estimating relevance levels of query-document pairs, retrievers and re-rankers usually adopt different models. Since retrievers need to recall relevant documents from a massive document pool, efficiency should be given priority. In that regard, traditional models such as BM25 (Robertson *et al.*, 1995) are used to construct initial retrievers. As to re-rankers, according to the stage wherein they play a role, re-rankers can be further categorized into early-stage re-rankers and latter-stage re-rankers. Compared to latter-stage re-rankers, early-stage re-ranker will focus more on efficiency but will pay more attention to effectiveness than retrievers. Since the number of documents considered by latter-stage re-rankers is small, latter-stage re-rankers will focus more on effectiveness. Conventional re-ranking models include learning to rank models (e.g., RankNet (Burges *et al.*, 2005b) and LambdaMart (Burges *et al.*, 2006)) and neural models (e.g., DRMM (Guo *et al.*, 2016b) and Duet (Mitra *et al.*, 2017)).

According to the number of re-rankers, the retrieval process can be defined in the following manner:

- **Single-stage Retrieval** ($n = 0$): the ranked list recalled by the initial retrieval is presented to search users without passing

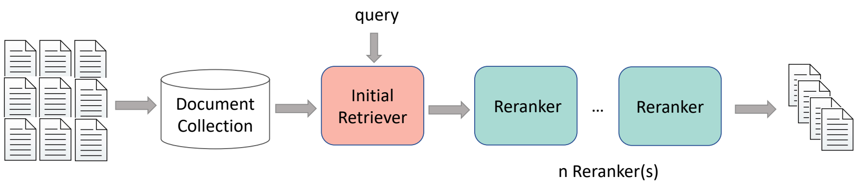

Figure 2.2: The retrieval architecture. According to the number of re-rankers, this retrieval process can defined as Single-stage Retrieval ($n = 0$), Two-stage Retrieval ($n = 1$) and Multi-stage Retrieval ($n \geq 2$).

through any re-ranker. This type of retrieval is applied in early retrieval frameworks such as boolean retrieval and scenarios in which the exact matching is sufficient and preferential.

- **Two-stage Retrieval ($n = 1$)**: besides the first-stage retrieval, existing IR frameworks also utilize a reranker to further improve the quality of the ranked list. Features that are not involved in the first-stage retrieval, such as multi-modal features, collected user behaviors and knowledge graphs, are also considered in the re-ranking stage.

- **Multi-stage Retrieval ($n \geq 2$)**: a multi-stage retrieval architecture comprises more than one reranking stage. Different re-rankers may adopt diverse structures and take advantage of different information sources. Common design can be that functions of each re-rankers are complement.

As to PTMs developed for IR, the trade-off between efficiency and effectiveness should also be considered according to stages for which these PTMs target. Especially, for the retrieval stage which focuses more on efficiency, PTMs are used to improve the performance of retrieval models (sparse, dense or hybrid). For example, ColBERT (Khattab and Zaharia, 2020) generates contextualized term embeddings for queries and documents with a BERT-based dual-encoder and executes two orders-of-magnitude faster per query compared to other baseline models. While for the re-ranking stage, PTMs need to delve with a small set of documents and capture more fine-grained relevance signals. For example, CEDR (MacAvaney *et al.*, 2019b) leverages the contextualized word embeddings of BERT to build a similarity matrix and then feed into an existing interaction-focused neural ranking model such as DRMM and KNRM. The [CLS] vector is also incorporated in CEDR to enhance the model's signals.

2.3 The System View of IR

As a practical system, the search system enables end users to perform IR tasks. Besides considering effectiveness and efficiency, a good search system should also be user-friendly. Hence, a good search system needs to deal with different issues existing in the real-world usage which require different components to cooperate. We depict the conventional framework of a search system in Figure 2.3. The search query issued by a user may be short, ambiguous and sometimes miss-spelling. In that regard, a query parser is needed to operate the original query and convert it to a query representation which can reveal the user's true intent to some extent. The operations on the original query may include rewriting, expansion and so on. From the document side, since different web documents have different page structures to organize the content, a document parser/encoder is then essential to process and index web pages. A document parser/encoder can also secure the speed in finding relevant documents for a search query. Without the document index, the search system would need to scan every document in the corpus, which requires considerable time and computing power. Besides the query parser and document parser/encoder, the retrieval & ranking component which is described above is used to provide most relevant results to the search user. In the framework of a search system, the core parts are data structure and storage which are considered in the document component. Delving into the history of the document index, we observe a paradigm shift from the symbolic search system to the neural search system. In the following, we briefly introduce how these two systems index documents and also their pros and cons.

- **Symbolic search system**: In a symbolic search system, rules are required to build the document parser which indexes, filters and sorts documents by a variety of criteria, and then translate

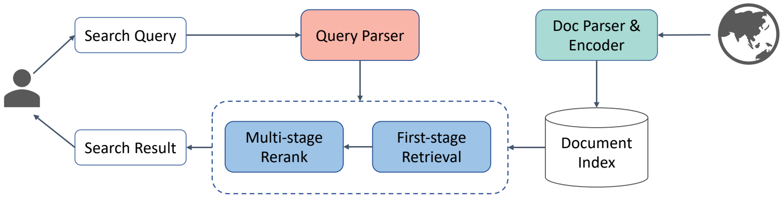

Figure 2.3: The framework of a practical search system.

this data into symbols that the system can understand. Hence the name, symbolic search. Especially, symbolic search system will index documents to build an inverted index which consists of two parts: a dictionary and postings. The dictionary contains all terms appeared in the document collection. Then for each term, a list that records which documents the term occurs in is generated. Each item in the list, which records that a term appeared in a document, is called a posting (or post). The list is conventionally called a posting list (or inverted list). The pros of symbolic search systems are the fast retrieval ability and the provided result is interpretable while the cons are that these systems are stuck using one language and require high maintenance cost.

- **Neural search system**: While the symbolic search system focuses more on "exact match", a neural search system attempts to capture "semantic match". Instead of designing a set of rules, the neural search system applies pre-trained models to obtain low-dimensional dense representations of documents, which develops a generalized ability of the search system to find relevant results. The document index in neural search systems is called vector index. Compared to symbolic search systems, neural search systems are more resilient to noise and easy to extend and scale which are the pros. The cons of neural search systems include less explainability and the need of lots of data for training.

After building the document index (inverted index or vector index), the search query and documents will be fed into retrieval and re-ranking stages which are elaborated in the above. In the retrieval and re-ranking stages, symbolic search systems prefer term-based models and learning

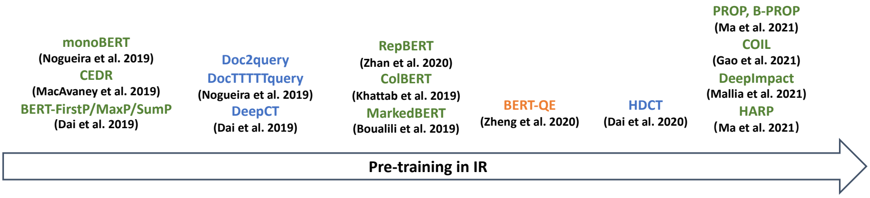

Figure 2.4: Recent PTMs in IR. "Orange", "Green" and "Blue" refer to the "Query Parser", "Retrieval and Rerank", and "Doc Parser & Encoder" stages for which PTMs target respectively.

to rank models while neural search systems adopt more dense retrieval models and neural ranking models. To note that, although the neural search method has become more widespread, symbolic search methods still play an important role in practical search systems nowadays. In fact, using a combination of neural search and symbolic search may result in optimal results.

Different PTMs are tailored for different components, i.e., "Query parser", "Doc Parser & Encoder" and "Retrieval and Rerank" in the search system. For example, BERT-QE (Zheng *et al.*, 2020) leverages BERT as the backbone network to expand queries and Mesh-BART (Chen and Lee, 2020) leverages user behavioral pattern such as clicks for generative query suggestion in the "Query Parser" component. DeepCT (Dai and Callan, 2019a) maps contextualized embeddings learned by BERT to term weights, then uses predicted term weights to replace the original TF field in the inverted index, which refines the "Doc Parser & Encoder" component. Compared to the "Query Parser" and "Doc Parser & Encoder" component, "Retrieval and Rerank" component receives much more attention in the sense that there exist lots of PTMs designed for this component. We show more recent examples in Figure 2.4 where different colors refer to different components on which these PTMs focus. Especially, "Orange" refers to the "Query Parser" component, "Green" refers to the "Retrieval and Rerank" component and "Blue" refers to the "Doc Parser & Encoder" component as shown in Figure 2.3.

3

Pre-training Methods Applied in Retrieval Component

In this section, we review existing works utilizing PTMs in the retrieval component.

- **Sparse Retrieval Models** (§ 3.1)
  - *Term Re-weighting* (§ 3.1.1)
  - *Document Expansion* (§ 3.1.2)
  - *Expansion + Re-weighting* (§ 3.1.3)
  - *Sparse Representation Learning* (§ 3.1.4)
- **Dense Retrieval Models** (§ 3.2)
  - *Basic Usage of Pre-training Methods* (§ 3.2.1)
  - *Pre-training Tailored for Dense Retrieval* (§ 3.2.2)
  - *Improved Fine-tuning Techniques* (§ 3.2.3)
  - *Advanced Topics* (§ 3.2.4)
- **Hybrid Retrieval Models** (§ 3.3)

Traditional search engines rely on term-based retrieval models like BM25 (Robertson and Zaragoza, 2009) for efficient retrieval. Recently, with the rapid progress in representation learning (Bengio *et al.*, 2013) and PTMs (Devlin *et al.*, 2018; Yang *et al.*, 2019b; Radford *et al.*, 2019), semantic retrieval (Cai *et al.*, 2021a), especially combined with pre-trained models, has become a popular paradigm to improve retrieval performance. They significantly outperform term-based retrieval models in terms of effectiveness and benefit downstream tasks (Yan *et al.*, 2021; Karpukhin *et al.*, 2020).

From the perspective of representation type and index mode, semantic retrieval models can be categorized into three classes: 1) **Sparse Retrieval Models:** improve retrieval by obtaining semantic-captured sparse representations and index them with the inverted index for efficient retrieval; 2) **Dense Retrieval Models:** project input texts (i.e., query and documents) into standalone dense representations and turn to approximate nearest neighbor (ANN) algorithms for fast retrieval; 3) **Hybrid Retrieval Models:** build sparse and dense retrieval models concurrently to absorb merits of both for better retrieval performance.

3.1 Sparse Retrieval Models

Sparse retrieval models focus on improving retrieval performance by either enhancing the bag-of-words (BoW) representations in classical term-based methods or mapping input texts into the "latent word" space. In this way, queries and documents are represented with sparse embeddings, where only a small number of dimensions are active. The sparse representation has attracted great attention as it can be easily integrated into the inverted index for efficient retrieval.

With the development of PTMs, pre-trained models have been widely employed to improve the capacity of sparse retrieval models. We summarize existing works that apply PTMs in sparse retrieval models into four classes, including term re-weighting, document expansion, expansion + re-weighting, and sparse representation learning.

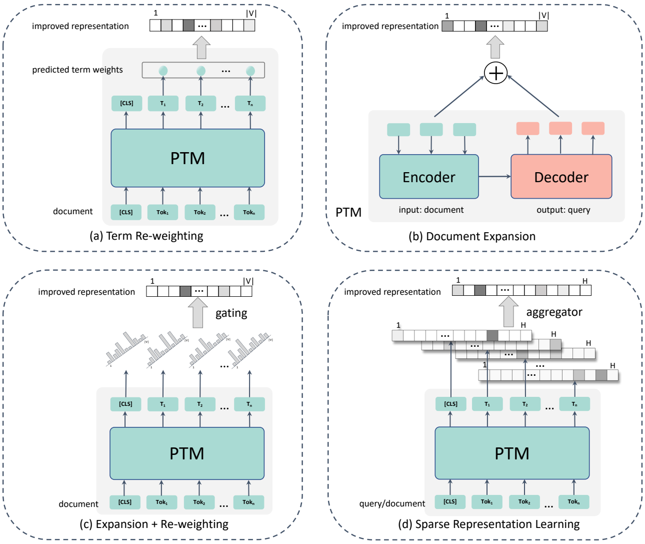

Figure 3.1: Four architectures of sparse retrieval models.

3.1.1 Term Re-weighting

One of the most direct way to improve the representation in term-based retrieval is to measure term weights with contextual semantics, instead of term frequency (Figure 3.1 (a)). There have been several works trying to utilize pre-trained models to estimate term weights. For example, Dai and Callan (2019a) and Dai and Callan (2020a) proposed a BERT-based framework (DeepCT) to evaluate the term importance in sentences/passages. It maps contextualized embeddings 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 the predicted weight could better estimate the term importance and improve term-based methods for the retrieval stage. Furthermore, Dai and Callan (2020b) introduced the HDCT model, which adapts DeepCT to estimate the term importance in long documents. It firstly estimates passage-level term weights as the DeepCT does. Then, these passage-level term weights are combined into document-level term weights through a weighted sum.

3.1.2 Document Expansion

In addition to explicitly predicting term weights, another kind of method is to augment the document with semantically related terms (Figure 3.1 (b)). In this way, on the one hand term weights of those elite terms can be promoted, and on the other hand, it can alleviate the vocabulary mismatch problem to some extent. Different from early methods that expand the original document by mining information from external resources (Sherman and Efron, 2017; Agirre *et al.*, 2010) or the collection itself (Efron *et al.*, 2012; Liu and Croft, 2004; Kurland and Lee, 2004), Nogueira *et al.* (2019a) fine-tuned a pre-trained language model T5 (Raffel *et al.*, 2019) with relevant query-document pairs. Then, the sequence-to-sequence (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, then the expanded document collection is indexed as usual. Finally, it relies on the BM25 algorithm to retrieve relevant documents. When combined with a re-ranking component, it achieves the state-of-the-art performance on MS MARCO (Nguyen *et al.*, 2016) and TREC CAR (Dietz *et al.*, 2017) retrieval benchmarks. Later, Yan *et al.* (2021) proposed a Unified Encoder-Decoder network (UED) to enhance document expansion with the document ranking task based on the assumption that document ranking and document expansion tasks share certain inherent relations and can benefit from each other. They firstly pre-trained the Transformer encoder-decoder architecture (Vaswani *et al.*, 2017). Concretely, the encoder is pre-trained to support document re-ranking with masked language modeling (MLM) (Devlin *et al.*, 2018) and sentence relation prediction (SRP) (Wang *et al.*, 2019b) objectives, and the decoder is pre-trained for query generation with the next sentence generation (NSG) task (Bi *et al.*, 2020a). Then, they made a joint fine-tuning process, where a mini-batch is selected with equal probability from the training data of document ranking or query generation tasks. Finally, the learned Seq2Seq model is used to expand documents as docTTTTTquery (Nogueira *et al.*, 2019a) does. Experimental results on two large-scale datasets show that UED achieves a new state-of-the-art performance on both MS MARCO Passage Ranking task (Nguyen *et al.*, 2016) and TREC 2019 Deep Learning Track (Craswell *et al.*, 2020). In fact, besides expanding documents, expanding queries is another way to improve retrieval performance. For example, Mao *et al.* (2020) proposed a novel query expansion method, named Generation-Augmented Retrieval (GAR). GAR augments semantics of a query with relevant contexts (expansion terms) through BART (Lewis *et al.*, 2019). They conducted the Seq2Seq learning with queries as inputs and various generation targets as outputs such as the answer, the sentence where the answer belongs to, and the title of a passage that contains the answer. Then, the generated contexts were appended to the query as the generation-augmented query for retrieval. The empirical results on the Natural Questions (Kwiatkowski *et al.*, 2019) and TriviaQA (Joshi *et al.*, 2017) datasets demonstrate that using multiple contexts from various generation targets is beneficial and yields better performance.

3.1.3 Expansion + Re-weighting

A more optimal method is to combine the idea of term re-weighting and document expansion, which learns the term importance in the whole vocabulary instead of existing tokens in the document (Figure 3.1 (c)). For example, Bai *et al.* (2020) proposed a novel framework SparTerm to build term-based sparse representations in the full vocabulary space. It takes BERT to map the frequency-based bag-of-words (BoW) representation to a sparse term importance distribution in the whole vocabulary. In this way, it can simultaneously learn the weight of existing terms and expand new terms for the document. To ensure the sparsity of final representations, SparTerm constructs a gating controller to generate binary and sparse signals across the dimension of vocabulary size. Results on the MS MARCO Passage Ranking task (Nguyen *et al.*, 2016) demonstrate that SparTerm outperforms all baselines, including DeepCT (Dai and Callan, 2020a) and docTTTTTquery (Nogueira *et al.*, 2019a). Later, Formal *et al.* (2021) proposed SPALDE to improve the SparTerm model (Bai *et al.*, 2020), which uses a saturate function to prevent some terms from dominating the representation and employs a *FLOPS* loss to enable end-to-end learning. In addition to doing the expansion and re-weighting simultaneously in a unified framework, Mallia *et al.* (2021) proposed a simple but effective model called DeepImpact, which leverages docTTTTTquery (Nogueira *et al.*, 2019a) to enrich the document collection firstly, and then uses BERT 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.

3.1.4 Sparse Representation Learning

Different from improving document representations in symbolic space, sparse representation learning methods focus on building sparse embeddings for queries and documents in the latent space, where each dimension of the latent space denotes a "latent word" (Figure 3.1 (d)). Then, the learned sparse representations can be stored and searched with the inverted index efficiently. For example, Jang *et al.* (2021) presented UHD-BERT, a novel sparse retrieval method based on pretrained language models. UHD-BERT empowers by extremely high dimensionality and controllable sparsity. It firstly obtains dense token embeddings for input texts (i.e., queries or documents) by BERT and maps them to high-dimensional vectors with a linear layer. Then, the *Winner-Take-All* mechanism is employed to remain top-k dimensions in the dense token embedding and get the sparse token embedding. Finally, it generates the sparse query/document representation by token-wise max pooling. Results showed that UHD-BERT outperforms previous sparse retrieval models significantly and delivers competitive performance compared to dense retrieval models. Recently, Yamada *et al.* (2021) introduced Binary Passage Retriever (BPR), which integrates the learning-to-hash technique into the state-of-the-art Dense Passage Retriever (DPR) (Karpukhin *et al.*, 2020) to represent input texts with compact binary codes. The BPR is learned using a multi-task objective that simultaneously trains the BERT-based dual-encoder and hash functions in an end-to-end manner. Based on the binary codes of queries and documents, relevant documents are retrieved using the Hamming distance. BPR drastically reduces the memory cost of the document index and obtains comparable accuracy on two standard open-domain

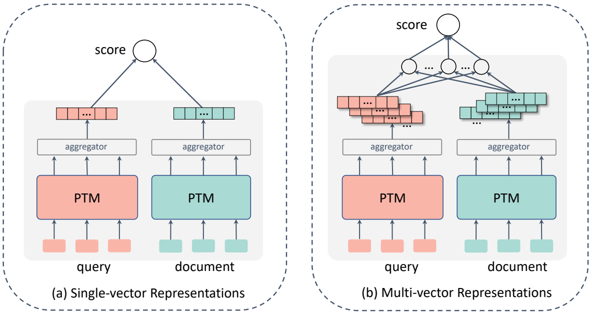

Figure 3.2: Basic architectures of dense retrieval models.

question answering benchmarks.

3.2 Dense Retrieval Models

Another research line, namely dense retrieval models, moves away from sparse representations to dense representations. Dense retrieval models employ the dual-encoder architecture, also called Siamese network (Bromley *et al.*, 1993), to learn low-dimensional dense embeddings for queries and documents. Afterward, the learned dense representations are indexed and searched via approximate nearest neighbor (ANN) algorithms to support online services.

Dense retrieval models usually consist of twin encoders that accept distinct inputs (i.e., queries and documents) and learn standalone dense embeddings for them independently. Then, a simple matching function (e.g., dot product or cosine similarity) is used to produce the final relevance score based on the learned query and document representations. In existing literatures, two dense retrieval families have emerged: single-vector representations (Figure 3.2 (a)), where the entire input text is represented by a single embedding, and multi-vector representations (Figure 3.2 (b)), where the input text is represented by multiple contextual embeddings.

3.2.1 Basic Usage of Pre-training Methods

One type of existing attempts at improving the quality of dense retrieval models focuses on finding more powerful representation learning functions. This is typically achieved by using a pre-trained language model as the encoder. In this way, the basic architecture of dense retrieval models can be formulated by:

$$rel(q, d) = f(E^{Q}_{PTM}(q), E^{D}_{PTM}(d)), \tag{3.1}$$

where E^{Q}_{PTM} and E^{D}_{PTM} are query and document encoders based on pre-trained models, and f is the similarity function.

One of representatives that apply pre-trained models for dense retrieval is DPR (Karpukhin *et al.*, 2020), which is proposed for OpenQA tasks. DPR models learns dense embeddings for text blocks with a BERT-based encoder, and queries are encoded with another independent BERT-based encoder. The OpenQA system based on DPR outperforms a strong Lucene BM25 system on a wide range of OpenQA datasets and is beneficial for end-to-end QA performance. For ad-hoc retrieval tasks, Zhan *et al.* (2020b) proposed the RepBERT model to replace BM25 for the retrieval stage. The model architecture of RepBERT is similar to DPR (Karpukhin *et al.*, 2020) except that RepBERT uses a shared BERT-based encoder for queries and documents. Then, inner products of query and document representations are regarded as relevance scores. Experimental results show that RepBERT outperforms BM25 on the MS MARCO Passage Ranking task (Nguyen *et al.*, 2016). Similarly, dense retrieval method also influences conversational search. For example, Yu *et al.* (2021) presented a Conversational Dense Retrieval system (ConvDR), which learns contextualized BERT embeddings for multi-turn conversational queries and documents respectively, and then retrieves documents solely using dot products between the obtained embeddings.

In addition to learning a single global representation for each query and each document, another more sophisticated approach is to obtain multiple vectors for queries and documents for matching. A state-of-the-art dense retriever, called ColBERT, is proposed by Khattab and Zaharia (2020). ColBERT generates contextualized term embeddings for queries and documents with a BERT-based dual-encoder, and then employs a term-based MaxSim operator to model fine-grained matching signals. 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. Results on MS MARCO (Nguyen *et al.*, 2016) and TREC CAR (Dietz *et al.*, 2017) 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.* (2021a), but the BERT-based query term embedding only interacts with the embedding of exactly matched document terms 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). Besides, an alternative way is to employ different encoders for queries and documents based on the motivation that documents are often lengthy and have diverse aspects, while queries are usually short and have focused topics. In this way, the document encoder abstracts the content into multiple embeddings—each embedding captures some aspects of the document, while the query encoder obtains a single embedding for each query (Luan *et al.*, 2020; Tang *et al.*, 2021). For example, Luan *et al.* (2020) proposed the 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 term embeddings at the top layer of BERT, 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 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 stronger performance than alternatives in open retrieval. Recently, Tang *et al.* (2021) proposed a novel multi-vectors representation method, which clusters BERT-based document term embeddings with k-means to generate multiple representations for each document. Experimental results on several popular ranking and QA datasets show that the model can boost retrieval results significantly and achieve state-of-the-art performance.

3.2.2 Pre-training Tailored for Dense Retrieval

At the beginning, the direct way to apply pre-training methods for dense retrieval is to use pre-trained models as encoders, and then fine-tune them on the target task, as Section 3.2.1 shows. Recently, several works have preliminarily explored pre-training techniques tailored for dense retrieval, e.g., additional pre-training stages with IR-oriented objectives, synthetic pre-training datasets, and new model architectures.

It has been generally accepted that the more similar the downstream task is to the pre-training task, the larger the gain (Zhang *et al.*, 2020; Ke *et al.*, 2020a). Researchers have therefore recognized the need to build new pre-training tasks that are better matched to retrieval. Lee *et al.* (2019b) is among the first to propose IR-oriented pre-training tasks. They treated sentences as pseudo-queries and matched them to the paragraph they originate from, namely inverse cloze task (ICT). They pre-trained the BERT-initialized dense retriever with ICT and then employed it to replace BM25 in OpenQA tasks. Later, Chang *et al.* (2019) combined ICT with the Body First Selection (BFS) and Wiki Link Prediction (WLP) tasks for ad-hoc retrieval, achieving significant improvement over baselines without pre-training or with other pre-training methods. Gao and Callan (2021c) recently used a unsupervised contrastive loss defined over the target search corpus to pre-train the passage embedding space in a query-agnostic fashion. In each of these cases, pre-training with new proposed tasks have shown to improve over their respective baselines. Besides, some studies explore the way to synthetize pre-training datasets for retrieval tasks (Reddy *et al.*, 2021; Ma *et al.*, 2021a; Liang *et al.*, 2020). For example, Reddy *et al.* (2021) trained a Seq2Seq generator using existing MRC data and used it to generate question-answer pairs from Wikipedia articles. They then created IR pre-training examples from these synthetic MRC examples. The experimental results show that pre-training on these generated examples improves the robustness of DPR (Karpukhin *et al.*, 2020).

Unlike previous works that design pre-training tasks or synthetize pre-training dataset, Gao and Callan (2021b) proposed a pre-training architecture for dense retrieval, called Condenser. Condenser performs language model pre-training actively conditioned on the [CLS] vector. As the result, it produces an information-rich [CLS] representation that can robustly condense an input sequence.

3.2.3 Improved Fine-tuning Techniques

In general, the dense retriever, which learns a single global representation based on the dual-encoder architecture and uses a simple similarity function for matching, is easy to learn and efficient for online services. However, their retrieval performance is always sub-optimal due to the limited modeling capability and rough fine-tuning paradigms. Recent researches have explored to design new fine-tuning techniques to boost the capacity of dense retrievers.

An approach to building a strong dense retriever is to distill the learned knowledge from a more complex model (e.g., a cross-encoder model) (Tahami *et al.*, 2020; Lin *et al.*, 2020a; Choi *et al.*, 2021; Hofstätter *et al.*, 2020). For example, Tahami *et al.* (2020) utilized knowledge distillation to compress the BERT-based cross-encoder network as a teacher model into the student dual-encoder model. This increases the prediction quality of BERT-based dense retrievers without affecting their inference speed. They evaluated the approach on three domain-popular datasets, and the proposed method achieved statistically significant gains. Besides, Lin *et al.* (2020a) distilled the knowledge from ColBERT's expressive MaxSim operator for computing relevance scores into a simple dot product function, 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. This approach improves query latency and greatly reduces the onerous storage requirement of ColBERT, while only making modest sacrifices in terms of effectiveness.

Another type of approaches to building a high-quality dense retriever focuses on learning better query/document representations using informative negative examples. In fact, the sampling strategy of negative examples is a crucial factor for determining the quality of the learned retrieval model. Generally, hard negative examples are viewed as informative negatives, because they can improve the model's ability to differentiate between similar examples. Thus, integrating hard negative mining strategies into the learning of dense retrievers is a widely concerned topic. One of the state-of-the-art methods is the Asynchronous ANCE training proposed by Xiong *et al.* (2020). They firstly warmed up the RoBERTa-based (Liu *et al.*, 2019) dense retriever with BM25 negatives, and then continued the ANCE training which periodically refreshes the ANN index and samples top-ranked documents as negatives. Experimental results indicate that ANCE elevates dense retrievers and convincingly surpasses baselines on several benchmarks. Later, Zhan *et al.* (2020a) and Zhan *et al.* (2021c) proposed a novel technique for dense retriever training, which constructs the document index based on a warmed-up dense retriever (e.g., ANCE (Xiong *et al.*, 2020) or STAR (Zhan *et al.*, 2021c)) beforehand. Then, at each training step, it performs full retrieval based on the fixed document index and updates the query encoder with top retrieved documents. Experiments on both passage ranking and document ranking tasks show that the proposed method significantly outperforms all competitive sparse and dense retrieval models. Recently, Hofstätter *et al.* (2021a) introduced a more efficient training method for dense retrievers, namely TAS-Balanced. The key insight of the proposed method is to train dense retrievers with TAS-Balanced batches, which composes training batches with topic-aware query sampling and margin-balanced negative sampling. They argued that the training batch in previous methods would be composed of random queries from the training set, leaving little information gain for in-batch negatives. By selecting queries from restrictive clusters, it leads to higher quality of retrieval results after the in-batch negative teaching.

3.2.4 Advanced Topics

Along with the development and achievement of dense retrieval models, researchers begin to explore more challenging but promising topics.

Some studies set about the end-to-end learning of dense retrievers and downstream tasks (e.g., machine reading comprehension (MRC)). For example, RAG (Lewis *et al.*, 2020) combines a pre-trained dual-encoder (DPR (Karpukhin *et al.*, 2020)) as the retriever with a pre-trained Seq2Seq model (BART (Lewis *et al.*, 2019)) as the generator for OpenQA tasks. The query encoder and the generator are fine-tuned end-to-end along with the fixed document encoder. The model evaluation on three OpenQA tasks demonstrates the state-of-the-art performance. Recently, Sachan *et al.* (2021) presented an end-to-end training method for retrieval-augmented OpenQA systems. They argued that such end-to-end training allows training signals to flow to the reader and then to the retriever better than staged-wise training. They built the EMDR^2 model, which initializes the dual-encoder retriever with BERT and builds the reader on top of T5. Experiments on three benchmark datasets demonstrate that the proposed method outperforms all existing approaches, achieving new state-of-the-art results.

Some researchers argue that existing dense retrieval models always separate two steps of representation learning and ANN index building (Zhang *et al.*, 2021; Zhan *et al.*, 2021a). This pattern suffers from a few drawbacks in practical scenarios. Firstly, the indexing process cannot benefit from supervised information because it uses the task-independent function to build the index. Besides, the representation and index are separately obtained and thus may not be optimally compatible. These problems all result in severely decayed retrieval performance. To address the problem, Zhan *et al.* (2021a) explored the joint training of the BERT-based dual-encoder and the PQ (Jegou *et al.*, 2010) index with a ranking-oriented loss. Experimental results on two retrieval benchmarks show that JPQ significantly outperforms existing baselines in terms of different trade-off settings.

Another topic is related to the negative sampling strategy. Generally, the construction of retrieval benchmark datasets often relies on a pooling process to recall a subset of documents for expert judging (Nguyen *et al.*, 2016; Craswell *et al.*, 2020). In this way, the training dataset only contains partially positive documents. During the learning of retrieval models, the rest of unlabeled documents are oftentimes assumed to be equally irrelevant (Zhan *et al.*, 2020b; Karpukhin *et al.*, 2020), which

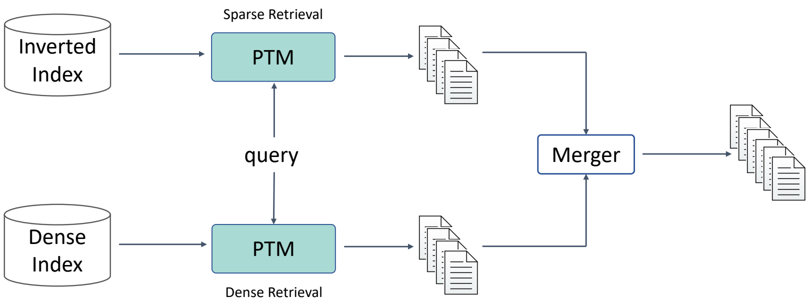

Figure 3.3: The architecture of hybrid retrieval models.

will lead to the well-known bias problem. To address the bias, it is necessary to devise smart learning strategies to achieve effective and efficient model training. For example, RocketQA (Ding *et al.*, 2020) and PAIR (Ren *et al.*, n.d.) used a stronger cross-encoder based on BERT to evaluate the relevance of unlabeled documents and remove those potentially false negatives. Recently, Prakash *et al.* (2021) improved upon ANCE (Xiong *et al.*, 2020) by estimating a negative sampling distribution based on a small but completely-labeled validation set for dense retrievers learning.

3.3 Hybrid Retrieval Models

Sparse retrieval models take a word or "latent word" as the unit of representations, which preserves strong discriminative power as the matching score is calculated based on exact matching signals. On the other hand, dense retrieval methods learn continuous embeddings for queries and documents to encode their semantic information, and the relevance is evaluated with soft matching signals. A natural approach to balance between the fidelity of sparse retrieval models and the generalization of dense retrieval models is to combine merits of them to build a hybrid retrieval model.

Hybrid retrieval models define multiple representation functions for queries/documents, and obtain their sparse and dense representations concurrently. Then, these representations are used to calculate the matching score with the defined merging method (Figure 3.3). To begin with, Seo *et al.* (2019) proposed DenSPI for the retrieval stage of OpenQA tasks. The DenseSPI model constructs the dense-sparse representation for each phrase in the document collection. The dense vector is represented as pointers to the start and end BERT-based token embeddings of the phrase, which is responsible for encoding syntactic or semantic information of the phrase with respect to its context. The sparse embedding uses bigram-based tf-idf for each phrase, which is good at encoding precise lexical information. During online services, there are three alternative merging ways to search, i.e., sparse-first search (SFS), dense-first search (DFS), and hybrid. Later, Lee *et al.* (2019a) proposed to learn contextual sparse representation for each phrase based on BERT to replace term-frequency-based sparse encodings in DenSPI (Seo *et al.*, 2019). This method leverages the 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. A more simple and direct way to build a hybrid retrieval model is to linearly combine matching scores of a sparse retrieval system and a dense retrieval system using a single trainable weight λ, tuned on the development set (Lin and Ma, 2021; Luan *et al.*, 2020). For example, Luan *et al.* (2020) proposed to linearly combine BM25 and ME-BERT (Luan *et al.*, 2020), which yields strong performance while maintaining the scalability. There are also works using more sophisticated merging methods. For example, Kuzi *et al.* (2020) performed an in-depth empirical analysis, which leverages a hybrid approach (BM25 + DE-BERT) with RM3 (Abdul-Jaleel *et al.*, 2004) as the merger for document retrieval. Results demonstrate the effectiveness of the hybrid approach and also shed some light on the complementary nature of the lexical and semantic models. Gao *et al.* (2020b) proposed the CLEAR model, which learns the BERT-based dense retriever with the residual of a sparse retrieval model (BM25). Experimental results show that retrieval from CLEAR without re-ranking is already almost as accurate as the BERT re-ranking pipeline.

4

Pre-training Methods Applied in Re-ranking Component

In this section, we review previous works applying PTMs in the re-ranking stage. Modern search engines take advantage of a multi-stage architecture in order to efficiently provide accurate result lists to users. In more detail, there can be a stack of complex re-rankers after the efficient first-stage retriever. The multi-stage cascaded architecture is very common and practical both in the industry (Yin *et al.*, 2016; Liu *et al.*, 2021d; Li and Xu, 2014) and the ranking leaderboard in the academia (Craswell *et al.*, 2021). Considering the large computational cost of Transformer-based pre-trained models, they are often employed to model the last stage re-ranker whose goal is to re-rank a small set of documents provided by previous stage. By learning contextualized representations and modeling complex interactions using deep Transformers for queries and documents, these pre-trained models achieved great success when applied to the re-ranking stage (Nogueira and Cho, 2019; Dai and Callan, 2019b; Lin *et al.*, 2021; MacAvaney *et al.*, 2019a; Nogueira *et al.*, 2019b; Yang *et al.*, 2019a).

There are mainly two schools of relevance modeling in IR: the discriminative modeling (Burges *et al.*, 2005a; Joachims, 2002; Liu, 2010) and the generative modeling (Lafferty and Zhai, 2003; Robertson

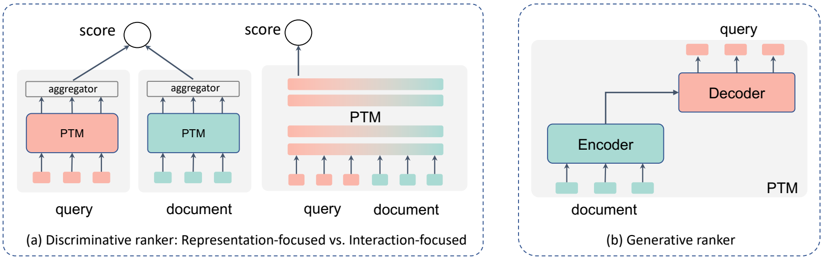

Figure 4.1: Two categories of retrieval model.

and Jones, 1976a; Ponte and Croft, 1998; Zhai, 2008). The discriminative modeling utilizes the strength of machine learning which can directly learn a "classifier" from labeled relevant judgments (Liu, 2010; Nogueira and Cho, 2019). For example, the neural ranking models can learn the distributional representations and the interactions between queries and documents, and then predict their relevance or rank order labels (Guo *et al.*, 2020c). But the generative modeling assumes that there is an underlying stochastic generative process between queries and documents, and aims to approximate the true relevance distribution. Obviously, there are two generation processes in IR: from documents to queries or from queries to documents. Statistical language models such as the query likelihood model consider the query generation process which ranks documents according to how likely query terms are generated from a document (Ponte and Croft, 1998; Zhai, 2008). Classic probabilistic relevance models such as Binary Independence Model are focused on describing how a document is generated from a given query (Lafferty and Zhai, 2003; Robertson and Jones, 1976a). On the other hand, according to the pre-training task (autoregressive vs. autoencoding (Yang *et al.*, 2019b)) and the model architecture (encoder-based vs. decoder-based/encoder-decoder), pre-trained models can also be classified into three categories, i.e., pre-trained models like BERT that mainly target discriminative tasks, pre-trained models like GPT that mainly target generative tasks and pre-trained models like T5 that can target both discriminative and generative tasks. We will illustrate how different pre-trained models can be applied to re-ranking using the above two schools of relevance modeling.

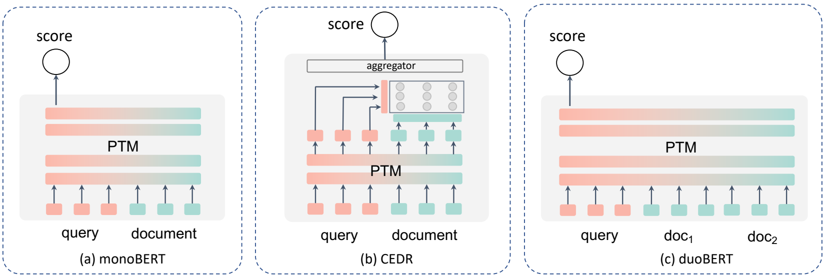

Figure 4.2: Three typical interaction-focused discriminative ranking models.

4.1 Discriminative ranking models

BERT and its successors have achieved great success in many discriminative tasks including sentence classification, sentence-pair classification tasks and so on (Devlin *et al.*, 2018; Liu *et al.*, 2019). These pre-trained models usually only use the encoder part of the Transformer and mainly target discriminative tasks. In the context of NeuIR literature (Guo *et al.*, 2020c), neural ranking models can be divided into two categories based on different assumptions over the features (extracted by the representation function or the interaction function) for relevance estimation, namely representation-focused architecture and interaction-focused architecture. Naturally, pre-trained models can also be applied in these two categories of architectures. In other words, we just replace neural ranking models with pre-trained models.

4.1.1 Basic usage: Representation-focused vs. Interaction-focused

As we mentioned above, pre-trained models can be used in two ways based on whether the relevance score is computed from comparing two separated encoded representations or their interactions.

Representation-focused: Representation-focused models (Figure 4.1 (a) right) usually adopt a bi-encoder architecture and learn dense representations of queries and documents independently, and then compare their representations to compute relevance via simple similarity functions such as cosine function or inner product. Without loss of generality, the representation-focused method could be abstracted by the following unified formulation:

$$relevance = f(PLM(Q), PLM(D)), \tag{4.1}$$

where Q and D denote query and document respectively, PLM is the pre-trained model such as BERT, and f is the evaluation function which computes the relevance score based on their representations. We take BERT as an example to explain the $PTM(Q/D)$ part, the format of the BERT input is usually "[CLS]+Q/D+[SEP]" and the output is always a 768-dimensional dense vector. Considering that there is a special classification token in BERT to aggregate the whole input sequence, i.e.,[CLS], researchers can use the output vector of [CLS] token to represent the input sequence. Also, we can do a mean/max pooling operation on the word embeddings of the whole input sequence to obtain a single 768-dimensional vector, and a weighted sum on multi-layers embeddings like ELMo is optional (Peters *et al.*, 2018a). Many similarity functions can be used to compute the relevance score such as cosine function, dot-product function, a feed-forward network and so on. For example, Qiao *et al.* (2019) encoded the query and document using BERT separately and used the [CLS] embedding of the last layer as their representations, and then calculates the ranking score via cosine similarity. Studies have shown that representation-based architectures are less effective than interaction-based architectures, but they are faster and can utilize the approximate nearest neighbor (ANN) search techniques to search from the pre-computed representations. So, recent studies usually apply representation-based architectures to the first-stage retrieval phase (see Section 3.2).

Interaction-focused: Interaction-focused architectures (Figure 4.1 (a) left) aim to capture the interaction between terms in the query and terms in the document and then produce the relevance score by abstract the interaction. The most straightforward applications of pre-trained models are to take them as interaction-focused architectures for the re-ranking stage. Since the powerful self-attention mechanism in Transformer can model the complex interactions between queries and documents. Moreover, the interaction cannot be pre-calculated until we see the input pair, which decides this type of model could only re-rank a small set of documents due to the large computational cost of computing all documents in the collection for each query. Without loss of generality, the interaction-focused method could also be abstracted by the following unified formulation:

$$relevance = f(PLM(Q, D)), \tag{4.2}$$

where f is the evaluation function which computes the relevance score based on the interaction. In this way, the input of BERT is a concatenation of query and document with special token separating: [CLS] + Q + [SEP] + D + [SEP]. And the output vector of [CLS] token can be seen as the interaction of the query and document, and then feed into a multi-layer feed-forward network which is the function f in Equation 4.2 to produce the relevance score. Following the learning to rank literature (Liu, 2010), the *pointwise* and *pairwise* learning objectives can also be adopted into fine-tuning pre-trained models. The *listwise* learning objectives are not applicable due to the input length limit of BERT-like pre-trained models is always 512.

MonoBERT (Nogueira and Cho, 2019) converts the ranking task into a text classification problem. MonoBERT takes the concatating query and document as input and feeds the [CLS] vector to a feed-forward network to obtain the relevance score of the passage being relevant. They use a *pointwise* loss function ,i.e., the cross-entropy loss, to fine-tune the BERT model. This simplest usage of BERT showed very effective for the ranking task in re-ranking stage. CEDR (MacAvaney *et al.*, 2019a) stacks a deep interaction model upon the monoBERT, that is, it leverages the contextualized word embeddings of BERT to build a similarity matrix and then feed into an existing interaction-focused neural ranking model such as DRMM (Guo *et al.*, 2016a) and KNRM (Xiong *et al.*, 2017a). The [CLS] vector is also incorporated in CEDR to enhance the model's signals. DuoBERT (Pradeep *et al.*, 2021) takes a sequence comprised of a query and two texts as input, i.e., [CLS]+Q+[SEP]+D_i+[SEP]+D_j+[SEP] and trained to estimate the candidate D_i is more relevant than D_j. Due to the length limit of BERT, the query, candidates D_i and D_j are truncated to 62, 223, and 223 tokens, respectively, so the entire sequence will have at most 512 tokens when concatenated with the [CLS] token and the three [SEP] tokens. Though it is more effective for passage ranking tasks than monoBERT,

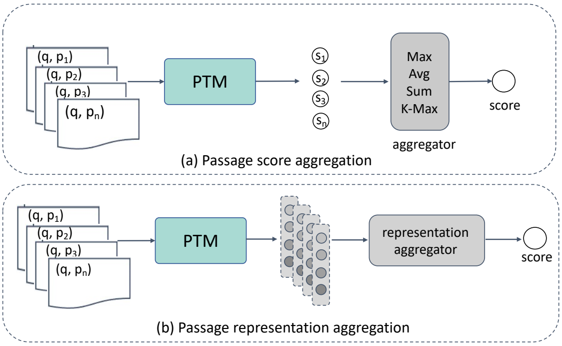

Figure 4.3: Two categories of passage aggregation methods.

when applying to document ranking tasks, the document has to truncate to meet the above length limit which will hurt the performance greatly.

4.1.2 Long document processing technique

Considering the length limit of Transformer-based (Vaswani *et al.*, 2017) pre-trained models is always 512, researchers have explored segmenting documents into passages and then do an aggregation to obtain the final document relevance score. Based on the aggregation type, the aggregation methods can be broadly categorized into two groups: *passage score aggregation* and *passage representation aggregation*.

Passage Score Aggregation: Dai and Callan (2019b) proposed to take the first passage score(BERT-firstP), the maximum passage score(BERT-maxP) or the sum of all passage scores(BERT-sumP) as the document relevance score. Documents are segmented into passages using a 150-word sliding window with a stride of 75 words in their experiments. The relevance judgments of segmented passages are consistent with the document, that is, if the document is relevant to a query, all the segmented passages are also relevant to the query and vise versa. However, according to the *Scope Hypothesis* (Robertson and Walker, 1994), the relevant parts could be in any position of a long document, and thus the document could be partially relevant to a query which will introduce noise if we treat all the passages as positive to the query. BERT-maxP and BERT-sumP perform better than BERT-firstP with the expense of the computational cost in their experiments since the first two methods need to train all the passage. IDCM (Hofstätter *et al.*, 2021b) firstly takes a lightweight and fast selection model namely ETM, e.g. Conv-KNRM (CK), to learn to select top-k passages. And it then takes a slow model like BERT namely ESM to estimate the passage-level relevance score independently and uses a fully-connected network to aggregate the top-k passage score. The IDCM framework consists of multiple non-differentiable operations (e.g., passage selection) that makes it difficult to use gradient descent-based methods for end-to-end optimization. Therefore, IDCM adopts an optimization pipeline with three training steps: (1) optimizing the ETM model for passage ranking; (2) extending the ETM optimization to full document ranking, and (3) optimizing the ESM model for passage selection using knowledge distillation. IDCM achieves comparable effectiveness to the full BERT-based ranker at lower computation cost and query latency. However, passage score aggregation methods use one passage to estimate the relevance one time, which will lose the long-range dependence.

Passage Representation Aggregation: Instead of only aggregating the passage score, aggregating the passage representation seems more promising in which the relevance score is estimated by considering all the passages together. PARADE (Li *et al.*, 2020a) splits a long text into a fixed number of fixed-length passages through padding and removing some passages, and then aggregate the [CLS] representation from each passage to produce the relevance score. They proposed two categories of aggregation method: using mathematical operation including the elmentwise mean, max and sum on the representation vectors; or using deep neural network including feed-forward networks, convolutional neural networks and Transformer layers. By aggregating the representations with more complicated architectures, $\text{PRADE}_{Transformer}$ can significantly improve the performance over passage score aggregation methods like BERT-maxP and other passage representation aggregation methods like PRADE_{max}. PCGM (Wu *et al.*, 2020) focuses on predicting the sequence of passage-level relevance judgments to avoid splitting a document into independent passages. It shows the superiority of capturing the context-aware fine-grained passage-level relevance signals. To be more specific, they firstly studied patterns of the passage-level information gain accumulate during a user's information seeking process and then they show the sequence of passage-level cumulative gain can be effectively predicted as a sequence prediction task. PCGM employs BERT to learn representations for query-passage pairs and aggregated passage representations with LSTM to obtain the passage cumulative gain. The model is trained on graded passage-level relevance judgments to predict the cumulative gain after each passage and the gain of a document's final passage is used as the document-level gain which is also the document-level relevance score. Experiments on TianGong-PDR dataset and the NTCIR-14 Web Chinese test collection show its effectiveness in improving ranking performance.

4.1.3 Model acceleration

The quadratic memory complexity of the self-attention models makes it difficult to fine-tune on long-text ranking tasks. Researches have explored several methods to reduce the high computational cost in the re-ranking stage including decoupling the interaction of query and document, model distillation, and dynamic modeling.

Decouple the Interaction: PreTTR (MacAvaney *et al.*, 2020) employed BERT model and proposed to decouple the low-level interaction of query and document into separate encoders and interact in the late BERT layers. Thus, the document representations can be pre-computed at indexing time and only the query needs to be inferenced online. So the most computational time comes from the interaction of the last few layers. When merging the query representation and document representation on layer 11 of BERT, PreTTR achieved a 42X speedup for TREC WebTrack while not significantly reducing the ranking performance. But merging at layer 11 performs poorly in terms of ranking effectiveness on the Robust04 dataset which indicates the ranking effectiveness of combining the representation of query and document at which layer depends on the dataset characteristics. When the query and document encoding is totally decoupled, it degrades to representation-focused architecture. Thus it's a trade-off between efficient and effective.

Model Distillation: Distilling a larger teacher model to a smaller student model is an effective way to reduce the computational cost in the deep learning literature (Hinton *et al.*, 2015; Sanh *et al.*, 2019). Gao *et al.* (2020a) investigated three methods to distill BERT for ranking including Ranker Distill, LM Distill + Fine-tuning, and LM Distill + Ranker Distill. Ranker Distill is to distill the BERT ranker behavior to a smaller BERT such as a 4-layer BERT or a 6-layer BERT. LM Distill aims to distill the general-purpose language knowledge learned from the BERT pre-training process via the language model predictions. LM Distill + Fine-tuning firstly runs a pre-trained BERT over a large text corpus and distills LM predictions from BERT to a smaller BERT, and then fine-tunes the distilled LM over the search task with the ranking objectives. LM Distill + Ranker Distill is to firstly distill LM to the distilled BERT and then distill the ranker behavior from a BERT ranker to the distilled BERT. Experiments on MS MARCO passage ranking task showed that distilling the ranker behavior alone is not sufficient and LM Distill + Ranker Distill method performs best across all datasets and different size of models. LM Distill + Fine-tuning method is able to reach original BERT ranker's effectiveness with a 6-layer distilled BERT ranker(2X speed-up), but fails with a 4-layer distilled BERT ranker(9X speed-up). On top of the TinyBERT model (Jiao *et al.*, 2020), Chen *et al.* (2021c) explored distill the student model with three other kinds of internal weights of the teacher model simultaneously in the fine-tuning stage, i.e., the attention weight, the hidden state weight, and the embedding weight. Experiments show that distilling more knowledge from the teacher model can better improve the ranking performance compared with the standard knowledge distillation.

Dynamic Modeling: Dynamic modeling which the model structures or parameters can be adapted to the input during inference is also another effective approach to reduce the computational cost (Han *et al.*, 2021b). Xin *et al.* (2020a) employed the idea of early exiting as in DeeBERT (Xin *et al.*, 2020b) to the document ranking task. Specially, extra classification layers are attached to transformer layers of a pre-trained BERT model and then fine-tune the model by simply minimizing the sum of loss functions of all classifiers. During inference, if the i^{th} layer classifier is confident of the prediction of the sample, early exiting is performed and subsequent transformer layers are skipped. The positive confidence threshold and the negative threshold are set to different values since they assume positive(relevant) documents need

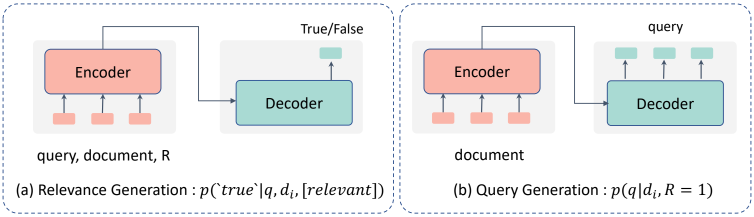

Figure 4.4: Two categories of generative ranking models.

more computations and the confidence score of positive documents is not only the exiting criterion but the score for re-ranking. Experiments on the MS MARCO passage ranking dataset show early exiting is able to accelerate inference by about 2.5X while maintaining the original model effectiveness. Cascade Transformer (Soldaini and Moschitti, 2020) takes each subset of Transformer layers as a re-ranker to drop a fixed proportion of samples after each "reranker". In the training phase, they uniformly sample one of the re-rankers to train, and update the layers below the selected re-ranker.

4.2 Generative ranking models

In addition to the discriminative ranking models, researchers have also explored applying the generative pre-trained language model like GPT (Radford *et al.*, 2019; Brown *et al.*, 2020a), BART (Lewis *et al.*, 2019) and T5 (Raffel *et al.*, 2020) to search. Based on the different generation process, these work can be categorized into *relevance generation* and *query generation*.

Relevance Generation: Relevance generation is focused on generating relevance labels by feeding the document and the query together to the generative pre-trained models, and the predicted logits of these relevance labels can be interpreted as relevance probabilities for ranking. Nogueira *et al.* (2020) proposed to utilize the sequence-to-sequence pre-trained language model T5 for document ranking. T5 introduces a unified framework that converts all text-based language problems into a text-to-text format, that is, it cast every task as feeding the model text as input and training it to generate some target text. For example, the sentence-pair classification task like QQP is done by feeding the sentence pair as input, e.g. "sentence1: sentece2:" and output "duplicate" token or "not_duplicate" token. To leverage T5 model, Nogueira et.al (Nogueira *et al.*, 2020) devised a text-to-text template for the ranking task to:

Input: Query: [Q] Document: [D] Relevant:, Output: true/false

So the model is fine-tuned to produce the target tokens "true" or "false" depending on whether the document is relevant or not to the query, and use the logit of the "true" token which is normalized with softmax function over the logits of "true" and "false" tokens as the document relevance score. Other target tokens like "yes/no" perform worse than the "true/false" tokens. Essentially, they proposed a method to exploit the T5's latent knowledge for the ranking tasks by "connecting" fine-tuned latent representations of relevance to output target tokens. T5-3B which firstly trained on MS MARCO passage ranking task and then transferred to other datasets outperform supervised training, i.e., Birch, BERT-maxP and PARADE, in a zero-shot manner.

Query Generation: Query generation is to formulate the ranking task as generating a query from a document's language model. This idea originated from the classical language model approach such as query likelihood model which is theoretically sound and very effective in traditional information retrieval literature (Ponte and Croft, 1998; Zhai, 2008; Lafferty and Zhai, 2001). Therefore, researchers have applied this idea to modern generative pre-trained models like GPT (Brown *et al.*, 2020a) and BART (Lewis *et al.*, 2019) for ranking task. Santos *et al.* (2020) leveraged the pre-trained language model such as GPT2 and BART to generate the query conditioned on document content, and the conditional likelihood is treated as the relevance score. The input format of the generative pre-trained models is as follows:

Input: <bos> passage, Output: <boq> question <eoq>

In order to leverage both positive and negative examples, they proposed two types of loss function: 1) unlikelihood loss on negative examples besides the likelihood on positive examples(LUL); 2) a pairwise ranking loss on the likelihood of positive and negative examples(RLL). They found that BART-base$_{RLL}$ is overall better than BART-base$_{LUL}$ and GPT2-base models. Experiments results showed the query generation approach using generative pre-trained models is as effective as state-of-the-art discriminative-based approaches for answer selection.

4.3 Hybrid models

Combining the generative and the discriminative modeling leads to the hybrid models. Liu et.al (Liu *et al.*, 2021a) proposed a multi-task learning approach to jointly learn discriminative and generative relevance modeling in one pre-trained language model. They assume that *joint discriminative and generative retrieval modeling leads to more generalized, and hence more effective retrieval models.* To verify this hypothesis, they use sequence-to-sequence pre-trained models like BART or encoder-only pre-trained models like BERT to learn discriminative ranking tasks and one or more supplementary language generation tasks, such as query generation task, questions generation task and anchor text generation task. For sequence-to-sequence pre-trained models, they fed the document and the query into encoder and decoder respectively. Then the query is generated in a sequence-to-sequence manner and the relevance score is computed by the last token of the entire sequence using a feedforward layer. Since the bidirectional attentions in BERT cannot fully adapt a seq2seq training strategy, they implemented a mix of bidirectional attention, unidirectional attention and cross attention mechanism to support sequence-to-sequence tasks by modifying the attention mask like the uniLM (Dong *et al.*, 2019). Experiments showed that joint learning discriminative ranking tasks and generative tasks lead to significant improvement on ranking task.

5

Pre-training Methods Applied in Other Components

In this section, we review existing work applying PTMs in the other components, such as query expansion, query rewriting, document summarization, snippet generation, etc. To elaborate, we divide these works into two categories and organize the following sections as:

- **Query Understanding**
  - *Query Expansion* (§ 5.1.1)
  - *Query Rewriting* (§ 5.1.2)
  - *Query Suggestion* (§ 5.1.3)
  - *Search Clarification* (§ 5.1.4)
  - *Personalized Search* (§ 5.1.5)
- **Document Summarization**
  - *Generic Document Summarization* (§ 5.2.1)
  - *Snippet Generation* (§ 5.2.2)
  - *Keyphrase Extraction* (§ 5.2.3)

5.1 Query Understanding

5.1.1 Query Expansion

Query expansion can be considered as an auxiliary task of document ranking, aiming to deal with the vocabulary mismatch problem or to mitigate the gap between queries and documents for better retrieval performance. To combine BERT embeddings with probabilistic language models, Naseri *et al.* (2021) developed an unsupervised contextualized query expansion model, namely CEQE, which expands existing queries based on keywords. Further experiments have demonstrated that CEQE can enhance retrieval effectiveness on multiple standard test collections. Besides, Padaki *et al.* (2020) proposed that query expansion should be tailored for models like BERT. Compared to keywords, feeding queries formatted in natural language into BERT-based models may achieve better reranking performance. In this regard, queries should be expanded with both a rich set of concepts and grammar structures that build word relations. An intuitive approach is to segment top-ranked documents of a specific query into text chunks and then rank these chunks (Zheng *et al.*, 2020; Zheng *et al.*, 2021). For example, Zheng *et al.* (2020) proposed BERT-QE which leverages BERT as the backbone network to expand queries through three phases: I) rerank candidate documents, II) select relevant text chunks for query expansion from the top-ranked documents, and III) rerank the selected expansion chunks. These chunks will then be concatenated with the original queries for scoring.

5.1.2 Query Rewriting

Query rewriting usually aims to 1) map long-tailed queries or questions into popular or frequent ones, 2) reformulate ambiguous input queries into well-formed queries to improve retrieval performance. To enhance conversational search, Lin *et al.* (2020b) utilized traditional IR query reformulation technique to realize historical query expansion (HQE) and then applied the T5-base (Raffel *et al.*, 2019) model for neural transfer reformulation (NTR), i.e., rewriting a raw utterance into a coreference-and-omission-free natural language question. There also exists a body of work towards matching user queries or questions to Frequently Asked Questions (FAQs) (Sakata *et al.*, 2019; Mass *et al.*, 2020; McCreery *et al.*, 2020). For instance, Mass *et al.* (2020) first employed BERT to calculate the semantic similarity between a query and the candidate FAQs. They further generated question candidates by fine-tuning GPT-2 (Radford *et al.*, 2019) in a well-designed unsupervised process and then filtered some noisy candidates according to the semantic similarity. Besides FAQ retrieval, query rewriting is also applied in spoken language understanding systems for friction reduction (Chen *et al.*, 2020b), or in dialogue systems to simplify the multi-turn dialogue (Liu *et al.*, 2021b). To reduce the requirement of high-quality query rewriting training pairs, Chen *et al.* (2020b) proposed a pre-training process which constructs more training objectives by making use of a large amount of readily available historical queries and their Natural Language Understanding (NLU) hypotheses (a serialized word sequence by concatenating domain, intent, slot type and the slot value).

5.1.3 Query Suggestion

As users' search intents become complex nowadays, a single query usually can not fulfill their information needs. In this regard, query/question suggestion techniques provide users with possible future query options, aiming to help users complete their search tasks with less effort in complex search scenarios, e.g., session search or conversational search. Compared to most previous methods (e.g., HRED-qs (Sordoni *et al.*, 2015), ACG (Dehghani *et al.*, 2017), and HSCM (Chen *et al.*, 2021a)) that used word2vec or GloVe vectors as an input to encode queries, Jiang and Wang (2018) constructed a heterogeneous session-flow graph on the AOL dataset and then applied the node2vec (Grover and Leskovec, 2016) tool to learn the term embeddings. The pre-trained term embeddings will then be fed into a reformulation inference network (RIN) to learn a session-level representation. RIN encodes historical reformulating actions with an RNN-based framework and achieves SOTA performances in both discriminative and generative query suggestion tasks. Some other methods have also attempted to employ Transformer-based models for query suggestion (Mustar *et al.*, 2020; Chen and Lee, 2020; Mitra *et al.*, 2020; Rosset *et al.*, 2020). For example, Chen and Lee (2020) proposed MeshBART which leverages user behavioral pattern such as clicks for generative query suggestion. To enhance conversational search, Rosset *et al.* (2020) focused on the usefulness of suggested questions and presented two novel systems. The first system, namely DeepSuggest, finetunes BERT to rank question candidates by jointly optimizing four learning objectives. The second one, DeepSuggest-NLG, adopts GPT-2 to generate question suggestions based on the maximum log-likelihood training. Their approaches leverage the weak supervision signals in the search process, grounding the suggestions to users' information-seeking trajectories and achieving significantly better performance in the usefulness evaluation. Besides user interactions, Mitra *et al.* (2020) also utilized search snippet text to recommend related questions in web search.

5.1.4 Search Clarification

As query suggestions are usually presented in a post-search manner, systems can also proactively ask users questions to clarify their information needs and reduce the uncertainty before returning the result list. Recently, search clarification has attracted much attention in various IR domains such as conversational search and dialogue systems. To begin with, Aliannejadi *et al.* (2019) proposed BERT-LeaQuR to encode both a query as well as its corresponding candidate questions and then employed a module called NeuQS to select high-quality clarifying questions. They also presented a new dataset named *Qulac* for conversational search, which collected clarifying questions through crowd-sourcing based on the ambiguous or faceted topics in the TREC Web track. Later, Hashemi *et al.* (2020) introduced Guided Transformer (GT), which utilizes external information sources such as top retrieved documents and clarifying questions to learn better representations of input sequences by optimizing a multi-task learning objective. Extensive experimental results on the *Qulac* dataset suggested that GT substantially outperforms strong baselines in both document retrieval and next clarifying question selection tasks. Besides, there are also researches focusing on ranking clarifying questions based on natural language inference (Kumar *et al.*, 2020) and user engagement prediction (Lotze *et al.*, 2021). Recently, Bi *et al.* (2021a) combined BERT with the maximum-marginal-relevance (MMR) criterion (Carbonell and Goldstein, 1998) to clarify user intents with fewer questions as possible. Their model, namely MMR-BERT, has shown promising efficacy in asking users clarifying questions on the *Qulac* dataset.

5.1.5 Personalized Search

Due to the variety of user propensity, search engines need to provide personalized search services by modeling individual preferences in appropriate scenarios. A common strategy for personalized search is encoding the search history to capture user's long-term and short-term interests. Aware of the remarkable learning power of the Transformer architecture, several recent studies have also focused on building frameworks for personalized search with some Transformer layers (Bi *et al.*, 2020b; Bi *et al.*, 2021b; Chen *et al.*, 2021a; Zhou *et al.*, 2020). For example, Zhou *et al.* (2021a) integrated transformer layers with Graph Attention Networks (GANs) and proposed a model named FNPS which considers both search behavior and friend network of users. To jointly optimizes session-level document re-ranking and query suggestion, Chen *et al.* (2021a) proposed a hybrid framework for session context modeling (HSCM) which leverages both intra-session and cross-session contextual information for personalization. Unlike general Web search, E-mail search requires personalization in conditions such as recency, user occupation, recipients, and attachments while protecting user privacy. To this end, Bi *et al.* (2021c) leveraged Transformer layers to encode personal e-mail search history, which only contains pre-processed features extracted from raw query and document text. As different features of one item should be emphasized in various search contexts, a fine-grained review-based transformer model RTM (Bi *et al.*, 2021b) was further proposed to enhance product search by dynamically encoding items at the review level. Experiment results have indicated both the efficacy of RTM in product search quality and its interpretability. Most existing personalized approaches do not involve a well-designed pre-training or self-supervised learning (SSL) process, merely utilizing the powerful learning ability of

Transformer-like architectures. Recently, some researchers focused on designing pre-training objectives for personalized search (Zhou *et al.*, 2021b) or session search (Zhu *et al.*, 2021). Their work have shown the great potential of applying contrastive learning in encoding user search history and the content.

5.2 Document Summarization

5.2.1 Generic Document Summarization

Generic document summarization aims at automatically compressing given documents into a piece of concise text while keeping salient information. The task is often generalized into two paradigms: *extractive summarization* and *abstractive summarization*. In extractive summarization, several sentences are selected from the original document and then concatenated to form a summary, while abstractive methods usually rewrite or paraphrase the document by language generation. Each paradigm has its own merits and limitations. For example, extractive summaries are more faithful in content, while they may also have low coherence or consistency between the selected sentences. Moreover, previous work shows that extractive approaches tend to choose long sentences. In contrast, abstractive summaries are more flexible while uncontrollable. Recently, PTMs have been proved effective to be applied in both extractive (Zhang *et al.*, 2019b; Liu, 2019; Zhong *et al.*, 2020; Wang *et al.*, 2019a; Xu *et al.*, 2020; Zhong *et al.*, 2019) and abstractive summarization (Zhang *et al.*, 2020; Dou *et al.*, 2020; Lewis *et al.*, 2019; Zou *et al.*, 2020; Saito *et al.*, 2020). Earlier, Yin and Pei (2015) built a strong CNN-based summarizer, namely DivSelect+CNNLM, to enhance extractive summarization by projecting sentences into dense distributed representations (*CNNLM*) and then constructing a diversified selection process (*DivSelect*). The CNNLM module is pre-trained on a large corpus and proved to learn better sentence representations by capturing more internal semantic features. Their method outperforms many traditional approaches such as LexRank (Erkan and Radev, 2004) and DivRank (Mei *et al.*, 2010) on the DUC 2002/2004 datasets, which can be considered as an early step in adapting PTMs in text summarization.

These years have witnessed the superb performance of PTMs such as BERT applied in various NLP tasks. Document summarization has also been greatly improved with the widespread use of these PTMs. For instance, Zhong *et al.* (2019) introduced BERT as external transferable knowledge (contextualized word embeddings) for extractive summarization and reported its superiority compared to word2vec (Mikolov *et al.*, 2013) and GloVe (Pennington *et al.*, 2014). Zhang *et al.* (2019a) first applied BERT into abstractive summarization via a two-stage decoding process: 1) firstly, generate the draft summary using a left-context-only decoder with copy mechanism; 2) then refine the summary using a refining decoder. Moreover, Liu (2019) and Liu and Lapata (2019) proposed a general framework called BERTSUM for both extractive summarization and abstractive summarization. Their experiments also indicated that the loss of the extractive task could further improve the abstractive task. To predict sentences instead of words, HIBERT (Zhang *et al.*, 2019b) maintains a hierarchical bidirectional transformer architecture and masks documents at sentence-level during encoding. As most work may cause a mismatch between the training objective and evaluation metrics by optimizing sentence-level ROUGE, Bae *et al.* (2019) presented a novel training signal that directly maximizes summary-level ROUGE scores through reinforcement learning. Their method can achieve better performance in the abstractive summarization task. To combine autoencoding and partially autoregressive language modeling tasks, Bao *et al.* (2020a) took Transformer as the backbone network to pre-train a unified language model UniLMv2. They designed a novel training procedure called pseudo-masked language model (PMLM) to jointly pre-train a bidirectional language model (LM) for language understanding and a sequence-to-sequence LM for language generation. Based on this technique, UniLMv2 performs better than other fine-tuning base-size pre-trained models such as BERTSUMABS and MASS (Song *et al.*, 2019).

While most approaches only involve pre-training tasks such as token or sentence masking, BART (Lewis *et al.*, 2019) corrupts raw text

The variants include BERTSUMEXT, BERTSUMABS, and BERTSUMEXTABS (multi-task learning).

with more noising functions (such as token deletion, text infilling, sentence permutation, and document rotation) and learns a model to reconstruct the original text. Therefore, BART is particularly effective when fine-tuned for abstractive summarization. It outperforms the best BERTSUM model by roughly 6.0 points on all ROUGE metrics in both CNN/DailyMail and XSum datasets. Unlike most previous approaches, MatchSUM takes extractive summarization as a semantic text matching problem and bypasses the difficulty of summary-level optimization by contrastive learning. The main point is that a good summary should be more semantically similar to the source document than the other candidates. Their approach borrows similar ideas from the IR domain and achieves considerable extractive summarization performance on six datasets. More elaborately, Google proposed a novel framework named PEGASUS (Zhang *et al.*, 2020), which adopts the gap-sentence generation (GSG) task tailored for abstractive summarization while pre-training. They hypothesized that using a pre-training objective that more closely resembles the downstream task leads to better and faster fine-tuning performance. To this end, gap sentences (indicates the most informational or important sentences within a document) will be selected and used as the target generation text for the remaining content. As a result, PEGASUS achieves SOTA performance in abstraction summarization on most mainstream public summarization datasets. Recently, some researchers also focused on I) improving the faithfulness of abstractive summaries by using saliency models or adding some guidances, i.e., CIT (Saito *et al.*, 2020) and GSum (Dou *et al.*, 2020), on II) distilling large pre-trained Transformers for summarization (Shleifer and Rush, 2020), or on III) legal domain related tasks (Huang *et al.*, 2021).

5.2.2 Snippet Generation

Different from generic document summarization, search snippets should highlight those points that are relevant in the context of a given query. Therefore, search snippet generation can be considered as one kind of Query-focused Summarization (QFS). Similar to generic document summarization, this body of work can also be divided into extractive approaches (Zhu *et al.*, 2019; Feigenblat *et al.*, 2017; Roitman *et al.*, 2020) and abstractive approaches (Laskar *et al.*, 2020a; Baumel *et al.*, 2018; Chen *et al.*, 2020a; Su *et al.*, 2020; Laskar *et al.*, 2020b). As some PTMs are proved to be effective in text generation, most existing work adopted PTMs to generate abstractive snippets. For instance, Laskar *et al.* (2020a) proposed a transfer learning technique with Transformer for the Query-Focused Abstractive Summarization (QFAS) task via a two-phase process. In the first phase, the BERTSUM (mentioned in Sec §5.2.1) model is pre-trained on a generic abstractive summarization corpus. They further fine-tuned the pre-trained model for the QFAS task on a target domain. During fine-tuning, they incorporated the query relevance by concatenating the query with the document as the input of the encoder. Baumel *et al.* (2018) presented RSA-QFS, which incorporates relevance-aware attention into a pre-trained sequence-to-sequence model (Nema *et al.*, 2017) for multi-document summarization. Despite that modern search engines usually present extractive snippets to search users, less effort has been made in employing PTMs for extractive snippet generation. One work may be (Zhu *et al.*, 2019), which developed a query-focused summarization model based on BERT to extract summaries from documents. Based on the model, they constructed massive query-focused summarization examples to enhance the modeling of query relevance and sentence context. One obstacle in query-focused document summarization is the lack of proper datasets. Some attention has also been paid on constructing benchmark datasets of certain scale for this task, e.g., *DUC* 2005-2007 QF-MDS task (Dang, 2005; Fisher and Roark, 2006), *Debatepedia* (IBM) (Nema *et al.*, 2017), *WikiRef* (Microsoft) (Zhu *et al.*, 2019), *qMDS* (Google) (Kulkarni *et al.*, 2020), etc. Besides retrieval systems, some other approaches (Su *et al.*, 2020; Savery *et al.*, 2020) are more suitable for Query-Answering (QA) system as they combine reading comprehension with language modeling.

5.2.3 Keyphrase Extraction

Keyphrase extraction or identification aims at extracting a set of informational, topical, and salient phrases from a document. It can not only provide users a quick view of result documents (similar to document summarization) but may also benefit downstream tasks such as document indexing, document recommendation, and query suggestion. Most of them formulated keyphrase extraction as a sequential labeling task (Lim *et al.*, 2020; Wu *et al.*, 2021; Park and Caragea, 2020; Sahrawat *et al.*, 2020; Liu *et al.*, 2020). For example, some work (Sahrawat *et al.*, 2020; Park and Caragea, 2020) adopted contextualized embeddings generated by BERT or SciBERT (Beltagy *et al.*, 2019) as the input of their BiLSTM-CRF architecture for scientific keyphrase extraction. Tang *et al.* (2019) used BERT with attention layer to automatically extract keywords from the clinical notes. From another perspective, Sun *et al.* (2020) proposed BERT-JointKPE which adopts multi-task learning to chunk self-contained phrases within a document and then rank these phrases by estimating their salience. Their method inherits the spirit of learning to rank and achieves promising keyphrase extraction performance in both the web and scientific domains.

6

Pre-training Methods Designed for IR

In this section, we introduce recent works designing PTMs tailored for IR (Lee *et al.*, 2019b; Chang *et al.*, 2019; Ma *et al.*, 2021b; Ma *et al.*, 2021c; Boualili *et al.*, 2020; Ma *et al.*, 2021d; Zou *et al.*, 2021; Gao and Callan, 2021a; Liu *et al.*, 2021d). General pre-trained models like BERT have achieved great success when applied to IR tasks on both the first-stage retrieval and the re-ranking stage. Beyond the direct application, researchers have explored PTMs with task-dependent self-supervised learning objectives (Zhang *et al.*, 2020; Glass *et al.*, 2020; Ke *et al.*, 2020b; Bao *et al.*, 2020b; Ma *et al.*, 2021b; Ma *et al.*, 2021c; Ma *et al.*, 2021d), and also model architectures designed for IR (MacAvaney *et al.*, 2020; Gao and Callan, 2021a). The underlying belief of task-dependent PTMs is that using a pre-training objective that more closely resembles the downstream task could lead to better fine-tuning performance with higher efficiency. Moreover, the task-oriented pre-trained models have the potential to provide the off-the-shelf ability for the target task which will considerly reduce the large cost of data labeling. This will be more valuable in IR since the most common setting is that of zero- or low-resource ad-hoc retrieval. But it will also may lose the one-fit-all ability of general pre-trained models, which means the task-oriented

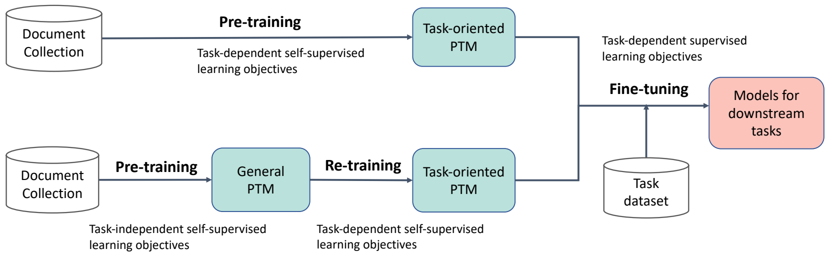

Figure 6.1: Two categories of pre-training paradigm.

pre-trained models may not perform well on other tasks. There is a trade-off between universality and specificity.

Boualili *et al.* (2020) proposed MarkedBERT which integrates exact match signals by marking the exact matched query-document terms with special tokens. Mask language modeling task (MLM) and next sentence prediction task (NSP) are used for pre-training MarkedBERT. The marking technique enhanced MLM can learn better text representations and teach the model to focus on exact matched terms which are useful for IR, but the NSP task which aims to learn inter-sentence coherence relationship diverges from IR requirement. Lee *et al.* (2019b) proposed Inverse Cloze Task (ICT) for passage retrieval in open domain question answering (openQA), which randomly samples a sentence from passage as pseudo query and takes the rest sentences as the document. Chang *et al.* (2019) proposed another two pre-training objectives besides ICT, i.e., Body First Selection (BFS) and Wiki Link Prediction (WLP). BFS randomly samples a sentence in the first section of a Wikipedia page as pseudo query and the document is a randomly sampled paragraph from the same page. WLP chooses a random sentence in the first section of a Wikipedia page as pseudo query, then a document is sampled from another page where there is a hyperlink between these two pages. These paragraph-level pre-training tasks are pre-trained on a bi-encoder architecture which can support the embedding-based retrieval. Experiments show that pre-training with these tasks can significantly outperform the widely used BM25 algorithm and the MLM pre-trained models. However, BFS and WLP heavily rely on the special structures of the document (e.g., multiple paragraph segmentation and hyperlinks), which hinders the method to be applied on a general text corpus.

When applied ICT pre-trained model to ad-hoc retrieval task, marginal gains observed on typical benchmark datasets (Ma *et al.*, 2021b). Ma *et al.* (2021b) proposed pre-training with representative words Prediction (PROP) for ad-hoc retrieval. The key idea of PROP is inspired by the traditional statistical language model for IR, specifically the query likelihood model (QL) (Ponte and Croft, 1998; Liu and Croft, 2005) proposed in the last century. Based on the Bayesian theorem, QL ranks the documents according to the query likelihood given the corresponding document language model. Inspired by QL, the Representative wOrds Prediction (ROP) task is proposed for pre-training. They assume *Representativeness can be a good proxy of Relevance* which means that if the language model could better identify the representative words of a document, it would better capture the relevance between a query and a document. Specifically, given an input document, they sample a pair of word sets according to the document language model which is defined by the multinomial unigram language model with Dirichlet prior smoothing (Zhai, 2008). The word set with higher likelihood is deemed as more "representative" of the document. Then they retrain BERT to predict the pairwise preference between the two sets of words, jointly with the Masked Language Model (MLM) objective. Despite its exciting performance on a variety of ad-hoc retrieval tasks, the effectiveness of PROP might be bounded by the classical unigram language model adopted in the ROP task construction process. To tackle this problem, B-PROP (Ma *et al.*, 2021c) leverages the powerful contextual language model BERT to replace the classical unigram language model for the ROP task construction, and re-train BERT itself towards the tailored objective for IR. Specifically, they proposed a novel contrastive method inspired by the divergence-from-randomness idea (Amati and Rijsbergen, 2002) to leverage BERT's attention mechanism to sample representative words from the document. The contrastive term distribution is obtained by computing the cross-entropy (i.e., the divergence) between a document term distribution and a random term distribution. Sampling from such contrastive term distribution, B-PROP obtain better representative words than the unigram language model and thus outperforms PROP on the downstream document ranking datasets.

Besides designing new pre-training tasks, some studies proposed to modify the Transformer model architecture to adapt IR tasks. For example, in the pre-training phase, Gao et.al proposed to add a short circuit from the lower layer of BERT (i.e., 6th layer for BERT-base) to the additional 2 layers put on top of BERT. And, the additional 2 layers received the token representations from the low layer instead of the former layer, and only the [CLS] token representation is from the former layer. They claim that by removing the burden of encoding local information and the syntactic structure of input text, the [CLS] token will focus more on the global meaning of the input text. In the fine-tuning phase, MacAvaney *et al.* (2020) proposed to block the attention flow between the query and the document at lower layers in a cross-encoder architecture. Thus, we can pre-compute the document representations and accelerate the inference for re-ranking. Here comes up the question, is Transformer the best architecture for IR?

In summary, researchers have explored two ways of PTMs tailored for IR, i.e. designing new pre-training tasks and new architectures. On one hand, designing pre-training tasks tailored for IR is based on the hypothesis that the pre-training task which more closely resembles the downstream task will lead to better fine-tuning performance. This hypothesis sounds reasonable and has also been verified to be effective in IR, however, it still doesn't touch the core of ranking, i.e., what contributes to relevance. More research is needed to understand what can truly benefit relevance modeling and how relevance can be modeled in pre-training. On the other hand, recent studies have made little progress on designing new architectures since existing works just slightly modify the attention mechanism in Transformer. There is still a long way to go for designing new architectures for IR which are theoretically well-justified and empirically effective.

7

Resources of Pre-training Methods in IR

In this section, we sort out some popular data repositories which are potential for the pre-training and fine-tuning process of PTMs in IR.

7.1 Datasets for pre-training

As shown in Figure 6.1, PTMs in IR requires a large-scale corpus with task-dependent self-supervised learning objectives. We consider the collections for pre-training tasks in IR with the following properties:

- **Large collection size**: In a broad sense, collection size is the necessity for pre-training tasks in any deep learning fields.
- **Structured documents**: The structures of a document include title, passages, sub-title, html structure, entity extractions and etc. These structures can be exploited in IR pre-training tasks to capture inter-page semantic relation. On the other hand, hyperlinks in the collection (e.g., anchor-page linking and page-page linking) provide intra-page semantic relation, which can also be used in IR pre-training.

Specifically, we consider the second property are not always necessary for IR pre-training tasks. But if a collection owns these properties, the

Dataset	Source	#Docs	Language	Latest crawl date
Books	Book	74M	ENG	2015
C4	web extracted text	0.3B	ENG	2019
Wikipedia	Wiki text	10M	Multi-lang	monthly update
RealNews	News	120GB	ENG	2019
Amazon	reviews	11GB	ENG	2003
WT10G	web pages	1.7M	ENG	1997
GOV2	pages in .GOV	25M	ENG	2004
CWP200T	chinese web pages	7B	CHN	2015
SogouT	Sogou web pages	1.17B	CHN	2016
ClueWeb09	web pages	1.04B	Multi-lang	2009
ClueWeb12	web pages	0.73B	ENG	2012
MS MARCO	Bing web pages	3.2M	ENG	2018

Table 7.1: Public available datasets which are potential for pre-training tasks.

https://github.com/huggingface/datasets/tree/master/datasets/bookcorpus
https://github.com/huggingface/datasets/tree/master/datasets/c4
https://dumps.wikimedia.org/
https://github.com/rowanz/grover/tree/master/realnews
https://snap.stanford.edu/data/web-Amazon.html
http://ir.dcs.gla.ac.uk/test_collections/wt10g.html
http://ir.dcs.gla.ac.uk/test_collections/access_to_data.html
http://www.sogou.com/labs/resource/t.php
https://lemurproject.org/clueweb09/
https://lemurproject.org/clueweb12/
https://microsoft.github.io/msmarco/

collection might be better for IR pre-training tasks. Given the suggested properties of IR pre-training dataset, we sort out some public available datasets which are potential for pre-training tasks, as shown in Table 7.1. According to the closeness to the IR, we categorize existing datasets into general text corpus and IR related corpus:

- **General text corpus**: The general text corpus is widely used in NLP researches for different tasks in different domains. These datasets generally contain a large amount of documents and provide implications for the classic pre-training tasks, e.g., masked language modeling (MLM) and next sentence prediction (NSP).
  - *Books*: Books are a rich source of both fine-grained information, how a character, an object or a scene looks like, as well as high-level semantics, what someone is thinking, feeling and how these states evolve through a story. This dataset provides rich descriptive explanations for visual content to align books to their movie releases.
  - *C4*: Colossal Clean Crawled Corpus (C4) is a dataset consisting of hundreds of gigabytes of clean English text scraped from the web, which can be used to pretrain language models and word representations.
  - *Wikipedia*: Wikipedia is a large-scale collection containing all Wikimedia wikis in the form of wikitext source and metadata embedded in XML. It takes advantages in perfect document structures, entity links, and rich information, which are suitable for pre-training tasks in IR.
  - *RealNews*: RealNews is a large corpus of news articles from Common Crawl. Data is scraped from Common Crawl, limited to the 5000 news domains indexed by Google News.News from Common Crawl dumps from December 2016 through March 2019 were used as training data; articles published in April 2019 from the April 2019 dump were used for evaluation.
  - *Amazon Reviews*: This dataset consists of reviews from amazon. The data span a period of 18 years, including 35 million

reviews up to March 2013. Reviews include product and user information, ratings, and a plaintext review.

- **IR related corpus**: Information retrieval aims to retrieve the relevant documents (items) from a large-scale corpus. The corpus can also be used for pre-training with the general pre-training objectives (e.g., MLM and NSP) and IR related pre-training objectives (Ma *et al.*, 2021b).
  - *WT10G*: WT10G (Web Track 10Gigabytes) was distributed by CSIRO in Australia (Chiang *et al.*, 2005). It is a crawl of web pages in 1997 and applied in many web-based experiments. The WT10G collection retains the properties of the 1997 web content which includes: the web link graph structure, server size distribution, inclusion of inter-domain links and inclusion of web pages on various subjects. The page content and hyperlinks in this dataset can be used in pre-training tasks by the methods discussed in Section 6.
  - *GOV2*: GOV2 is a crawl of .gov sites in the early of 2004 which includes html and text, plus the extracted text of pdf, word and postscript. The collection is 426GB in size and contains 25 million documents. The large proportion of web pages is potential for pre-training tasks with text-based self-supervised learning objectives.
  - *CWP200T, SogouT*: CWP200T and SogouT (Luo *et al.*, 2017) are the web page collections in chinese, which are provided by China Computer Federation (CCF) and Sogou search engine, respectively. Both collections are suitable for pre-training tasks in chinese IR.
  - *Clubweb*: Clueweb is a large-scale web document collection provided by CMU. The full collection of Clueweb09 contains about 1 billion web pages in ten languages that were collected in January and February 2009. Clueweb12 was further created based Clueweb09 with several data cleaning strategies. Both datasets are widely used in IR and several tracks of the TREC conference.

  - *MS MARCO*: MS MARCO (Craswell *et al.*, 2021) is a popular large-scale document collection which contains about 3.2 million available documents, which are from the Bing search engine. Besides, 1 million non-question queries are also included in this dataset for different retrieval tasks.

For general text corpus, we believe there are a number of corpus which is not listed in our paper. We recommend readers to this link to further explore the available datasets for pre-training tasks. And, the corpus with web pages mostly contain inter-document (e.g., html structure) and intra-document (e.g., hyperlinks, anchor-page links) relations, which can be used to build text pairs for IR pre-training tasks.

7.2 Datasets for fine-tuning

We sort out some datasets for downstream fine-tuning tasks. These tasks are categorized into document-oriented tasks, query-oriented tasks and others as follows:

- **Document-oriented**
  - *First stage retrieval* (**FSR**): retrieval stage from the full collection.
  - *Ad-hoc ranking* (**AR**): Ranking a candidate list given a query.
  - *Session search* (**SS**): Ranking a candidate list given a query and historical interactions.
  - *Multi-modal ranking* (**MMR**): Given a query, rank the candidate list where each item contains multiple heterogeneous information such as text, picture and html structure.
  - *Personalized Search* (**PS**): User-specific Ranking.
- **Query-oriented**

https://github.com/huggingface/datasets/tree/master/datasets

  - *Query reformulation* (**QR**): The process of iteratively modifying a query to improve the quality of a search engine results in order to satisfy user's information need.
  - *Query Suggestion* (**QS**): Providing a suggestion which may be a reformulated query to better represent a user's search intent.
  - *Query Clarification* (**QC**): Identifying user's search intent during a session.

- **Others**
  - *Document Summarization* (**DS**): The process of shortening a document to create a subset (a summary) that represents the most important information in this document.
  - *Snippet Generation* (**SG**): Query-specific document summarization.
  - *Keyphrase Extraction* (**KE**): It is also known as Keyword Extraction, which aims to automatically extracts the most used and most important terms in a document.

The detailed description of each collections are as follows:

1. Robust track is a classic ad-hoc retrieval task in TREC which focuses on poorly performing topics. The released annotated collection only includes 250 queries and 50 queries in Robust04 and Robust05, respectively. This collection is used for evaluation in most experimental settings.

2. TREC Million Query (MQ) Track conducts an ad-hoc retrieval task over a large set of queries and a large collection of documents. The final released dataset contains a four-level relevance judgement for each query-document pair.

3. Clueweb is another large-scale web search dataset provided by CMU. The "Category B" data set consists of the first English pages, which is roughly the first 50 million pages of the entire data set.

Dataset	Subdata	Size	Source	Potential tasks
Robust	Robust04 Robust05	0.5M docs, 250 queries 1M docs, 50 queries	TREC Robust Track	FSR, AR, QR
TREC MQ	MQ2007 MQ2008	6.5K docs, 1.7K queries 1.4K docs, 784 queries	TREC Million Query track	FSR, AR, QR
Clueweb	09-CatB 12-CatB	50M docs, 150 queries 50M docs	Web pages	FSR, AR, QR, KE
TREC web track	99-2014	See	TREC web track	FSR, AR, QR
TREC DL track	2019-2021	See	TREC Deep Learning track	FSR, AR
AOL	\	6M queries	AOL Query logs	AR,SS,PS,QR,QS
Sogou-QCL	\	9M docs, 0.5M queries	Sogou Query logs	AR, QR
Sogou-SRR	\	63K results, 6K queries	Sogou Query logs	AR, MMR, QR
Tiangong-ST	\	0.3M docs, 40K queries	Sogou Query logs	AR, SS, QR, QS
Qulac	\	10K question-answer pairs	TREC Web Track	AR, QR, QC
BEIR	7 IR tasks	Vary from tasks	Wiki, Quaro, Twittter, News and etc.	FSR,AR, etc.
MS MARCO	2019-20	1M queries, 8.8M passages, 3.2M docs	TREC Deep Learning Track	FSR, AR, QR
TREC CAR	\	30M paras, 2M queries	TREC Complex answer retrieval	AR, QR, KE
CNN / Daily Mail	\	0.3M docs	Human generated abstracts	DS
New York Times(NYT)	\	20K docs	News articles	DS
Debatepedia	\	1,303 debates	Debate key points	SG, DS
DUC	2001-07	300 clusters, See	Doc understanding conference	SG, DS
WIKIREF	\	0.3M samples	QFS benchmark	SG, DS

Table 7.2: Datasets for different downstream tasks in IR. Abbreviations in potential tasks are detailed in Section 7.2.

https://trec.nist.gov/data/webmain.html
https://microsoft.github.io/msmarco/TREC-Deep-Learning.html
https://duc.nist.gov/data.html

4. TREC web track exploits the documents from Clueweb. The goal is to explore and evaluate specific aspects of Web retrieval, including traditional ad-hoc retrieval task, risk-sensitive task and diversity search task.

5. TREC Deep Learning Track studies IR in a large training data regime. It contains two tasks: Passage ranking and document ranking; Two subtasks are included in each case: full ranking and reranking. Researchers usually take this dataset as an evaluation set by training a retrieval model on a large-scale dataset such as MSMARCO.

6. AOL is a public available query log released by the internet company AOL. The collection contains the query session, anonymized user ids and clicked documents, which are suitable for ad-hoc ranking, session search ranking, personalized search ranking, query reformulation and suggestion.

7. Sogou-QCL, Sogou-SRR (Search Result Relevance) and Tiangong-ST dataset were created from Sougou search engine to support research on IR. The Sogou-QCL collection consists of 537,366 queries, more than 9 million Chinese web pages, and five kinds of relevance labels assessed by click models. Meanwhile, the dataset also includes 2,000 queries with 4-level human assessed relevance labels.

8. The Sogou-SRR dataset consists of 6,338 queries and corresponding top 10 search results. Each search result contains the screenshot, title, snippet, HTML source code, parse tree, url as well as a 4-grade relevance score (1-4) and the result type. The heterogeneous information provides opportunity for multi-modal ranking.

9. Tiangong-ST provides 147,155 refined Web search sessions, 40,596 unique queries, 297,597 Web pages, and six kinds of weak relevance labels assessed by click models. Different from Sogou-QCL and Sogou-SRR, the session information provided in this dataset is able to be used in session search ranking.

10. Qulac was collected through crowdsourcing based on the topics in the TREC Web Track 2009-2012. It is a dataset on asking Questions for Lack of Clarity in open-domain information-seeking conversations.

11. The dataset contains 198 topics where each topic has recognized as either "ambiguous" or "faceted". The clarifying questions are collected through crowdsourcing for each topic.Based on each topic-facet pair, the answers to each clarifying question are collected. The average facets per topic is 3.85 ± 1.05. The facets and topics in this collection can be used for query clarification task.

12. BEIR (Benchmarking IR) (Thakur *et al.*, 2021) is a new heterogeneous benchmark containing different IR tasks. The benchmark contains a total of 9 IR tasks (Fact Checking, Citation Prediction, Duplicate Question Retrieval, Argument Retrieval, News Retrieval, Question Answering, Tweet Retrieval, Biomedical IR, Entity Retrieval) from 17 different datasets. Through BEIR, it is possible to systematically study the zero-shot generalization capabilities of multiple neural retrieval approaches.

13. MS MARCO (Craswell *et al.*, 2021) is a popular large-scale document collection which contains about 3.2 million available documents, which are from the Bing search engine. Besides, 1 million non-question queries are also included in this dataset for different retrieval tasks.

14. The TREC Complex Answer Retrieval(CAR) track uses topics, outlines, and paragraphs that are extracted from English Wikipedia. Wikipedia articles are split into the outline of sections and the contained paragraphs. The complex topics are chosen from articles on open information needs, i.e., not people, not organizations, not events, etc. It contains a passage task and an entity Task, where the latter can be used in keyphrase extraction tasks.

15. The CNN/Daily Mail dataset is a large collection of news articles and modified for summarization. It consists of more than 280,000 training samples and 11490 test set samples.

16. New York Times(NYT) consists of many news articles for summarization.

17. Debatepedia is collected from *debatepedia.org*. It is an encyclopedia of pro and con arguments and quotes on critical debate topics. There are 663 debates in the corpus, which belong to 53 overlapping categories such as Politics, Law, Crime, Environment, Health, Morality, Religion, etc. The average number of queries per debate is 5 and the average number of documents per query is 4.

18. The DUC dataset is a dataset for document summarization. In most experiments, it is used for testing only. It consists of 500 news articles, each paired with four human written summaries. In DUC2004, it consists of 50 clusters of Text REtrieval Conference (TREC) documents, from the following collections: AP newswire, 1998-2000; New York Times newswire, 1998-2000; Xinhua News Agency (English version), 1996-2000. Each cluster contained on average 10 documents. For the details of other versions, please refer to here .

19. WIKIREF is a large query-focused summarization dataset from Wikipedia which aims to generate summarization with a given query. It contains more than 280,000 examples.

7.3 leaderboards

In this section, we list several public leaderboards for researchers to understand the state-of-the-art methods in different tasks.

1. MS MARCO (Passage retrieval and document retrieval task): https://microsoft.github.io/msmarco/

2. DuReader (Machine Reading Comprehension task):https://ai.baidu.com/broad/leaderboard?dataset=dureader

---

https://duc.nist.gov/data.html

3. Robust04 (Document retrieval task): https://paperswithcode.com/sota/ad-hoc-information-retrieval-on-trec-robust04

4. CNN/Mail (Documents summarization task): https://paperswithcode.com/sota/document-summarization-on-cnn-daily-mail

5. Baidu DuIE (Entity extraction task): https://ai.baidu.com/broad/leaderboard?dataset=dureader

6. Benchmarking IR (BEIR) (Passage retrieval and document retrieval task): https://github.com/UKPLab/beir

8

Challenge and Future Work

In this section, we discuss current challenges and suggests some future directions for PTMs in IR. Some of these topics are important but have not been well addressed in this field, while some are very promising directions for future research.

8.1 New Objectives & Architectures Tailored for IR

Although the general-purpose PTMs are always our pursuits for learning the universal knowledge of languages, designing the pre-training and tuning protocol that more closely resemble downstream tasks is admittedly a more efficient way to obtain better performance on specific tasks (Zhang *et al.*, 2020; Ke *et al.*, 2020a). From the aspect of pre-training objectives, pre-training model architectures, and model tuning methods for IR, there have been some preliminary works, but we believe it deserves further exploring towards these directions.

New Pre-Training Objectives. As described in Section 6, there have been some pioneer studies (Lee *et al.*, 2019b; Chang *et al.*, 2019; Guu *et al.*, 2020; Ma *et al.*, 2021b; Ma *et al.*, 2021c; Liu *et al.*, 2021e; Ma *et al.*, 2021d) on the pre-training objectives tailored for IR. For example, Lee *et al.* (2019b) proposed to pre-train on a large-scale corpus with the

Inverse Cloze Task (ICT) for passage retrieval, which randomly samples a sentence from passage as pseudo query and takes the rest sentences as the document. Besides ICT, Chang *et al.* (2019) also proposed the Body First Selection (BFS) and Wiki Link Prediction (WLP) to capture the inner-page and inter-page semantic relations for passage retrieval in QA scenario. For the ranking stage, Ma *et al.* (2021b) and Ma *et al.* (2021c) proposed to pre-train with the Representative Words Prediction (ROP) objective for ad-hoc retrieval, which achieves significant improvement over baselines without pre-training or with other pre-training methods. In addition to constructing pseudo query-document pairs from the raw textual corpus, some researches turned to relying on certain corpus structures. For example, Ma *et al.* (2021d) proposed to leverage the large-scale hyperlinks and anchor texts to pre-train the language model for ad-hoc retrieval. Experimental results show that pre-training with four objectives based on the hyperlinks (i.e., RQP, QDM, RDP, and ACM) and the MLM objective jointly achieves state-of-the-art performance on two large-scale ad-hoc retrieval datasets. On the whole, the underlying idea of all these pre-training objectives tailored for IR is to resemble the relevance relationship between natural language queries and documents. However, it is still in its infancy stage to design suitable pre-training objectives for IR.

New Architectures. Beyond designing new pre-training tasks for IR, another research line is to design novel architectures according to specific downstream tasks. For example, towards the bi-encoder models for dense retrieval, Gao and Callan (2021b) argued that language models like BERT have a non-optimal attention structure to aggregate sophisticated information into a single dense representation for retrieval. Based on these observations, they introduced a novel Transformer pre-training architecture, Condenser, to address structural readiness during pre-training. Experimental results show that Condenser yields sizable improvement over standard LM and shows comparable performance to strong task-specific PTMs. Similarly, in order to obtain better text sequence embeddings for dense retrieval, Lu *et al.* (2021) presented a new auto-encoder architecture with restricted attention flexibility. In this way, the new architecture could create an information bottleneck in the auto-encoder and force the encoder to provide better text representations. Experimental results confirm that the seed dense retrieval model produces higher accuracy and better few-shot ability. However, compared with attempts investigating new pre-training objectives for IR, designing ingenious model architecture which is suitable for IR pre-training has not been well explored.

Beyond Fine-Tuning. Currently, fine-tuning is the dominant method to transfer the knowledge of PTMs to downstream tasks, but it has some undesired limitations: (1) it performs poorly on some downstream tasks without enough supervision data to support fine-tuning; (2) it is inefficient to fine-tune parameters on every downstream task. Recently, with the emerging of GPT-3 (Brown *et al.*, 2020a), a novel genre for model tuning, namely prompt tuning (Liu *et al.*, 2021e), is attracting more and more attention. By designing, generating and searching discrete (Petroni *et al.*, 2019; Gao *et al.*, 2020c) or continuous (Liu *et al.*, 2021c; Lester *et al.*, 2021) prompts for specific downstream tasks, these models could: (1) bridge the gap between pre-training and fine-tuning, and thereby perform better on downstream tasks; (2) reduce the computational cost on fine-tuning the tremendous amounts of parameters. To sum up, prompt tuning is a promising way to stimulate the linguistic and world knowledge distributed in PTMs. In fact, it has achieved exciting results in some fields, such as information extraction (Chen *et al.*, 2021b; Han *et al.*, 2021a), text classification (Puri and Catanzaro, 2019; Schick and Schütze, 2020), and fact probing (Petroni *et al.*, 2019; Jiang *et al.*, 2020). However, there has been no mature work on prompt tuning in the field of IR. From another perspective, the design of most of existing PTMs has been driven by the results on the fine-tuning paradigm, but it is not clear whether an exploration of different PTMs could lead to pre-trained models that are more effective when used with prompt tuning to solve IR tasks.

8.2 Utilizing Multi-Source Data for Pre-training in IR

Developing PTMs that take advantage of multi-source heterogeneous data, including multi-lingual, multi-modal, and external knowledge, for IR is another promising direction. On one hand, abundant data resources are vital significance for pre-training, and on the other hand, incorporating extra data has great potential to enhance text representations for IR tasks.

Multi-modal Pre-Training for IR. Large-scale pre-training and different downstream applications have been widely developed with diverse real-world modalities, such as audio, video, image, and text. In recent years, there has been an upsurging interest in cross-modal tasks, e.g., image-text retrieval (Lee *et al.*, 2018b; Huo *et al.*, 2021), visual question answering (Alberti *et al.*, 2019; Antol *et al.*, 2015), and image caption (Vinyals *et al.*, 2015; Johnson *et al.*, 2016). Meanwhile, large-scale PTMs also have enhanced research interests in the intersection of multiple modalities, such as image-text (Lu *et al.*, 2019; Li *et al.*, 2019), video-text (Sun *et al.*, 2019a), or audio-text (Chuang *et al.*, 2019). Among the Vision-and-Language pre-training (VLP) research, most current works focus on the interaction of images and texts (Li *et al.*, 2019; Su *et al.*, 2019; Lu *et al.*, 2019; Li *et al.*, 2020c), expecting to have a joint understanding of both to improve the performance on single-modal and multi-modal tasks. Since 2019, many VLP models have been proposed and achieved great success for various downstream tasks. Specially, Cao *et al.* (2020) probed the pre-trained Vision-Language models over nine tasks defined in SentEval (Conneau and Kiela, 2018). Results show that the pre-trained model indeed encodes richer linguistic knowledge to enhance NLP tasks. Similarly, the unified-modal pre-training architecture, namely UNIMO, proposed by Li *et al.* (2020b) shows improved performance on single-modal NLP tasks by allowing textual knowledge and visual knowledge to enhance each other in the unified semantic space. However, most of these works are not evaluated on IR tasks. Besides, although multi-modal PTMs have been witnessed numerous advances in the last two years, Cao *et al.* (2020) proved that the textual modality is more dominant than image in the multi-modal pre-training. Based on this, the profit of cross-modal learning is mainly reflected on image-based tasks. Thus, designing better vision-language pre-training objectives pointing at text-based IR tasks deserves further exploration. On the other hand, utilizing more modalities (e.g., audio or video) and more data is another problem that needs to be further explored in the future.

Multi-lingual Pre-Training for IR. Despite the rapid progress in PTMs, most prior work has been exclusively on English, where large-scale annotations are readily accessible. However, we live in a multi-lingual world, and training a large language model for each language is not an elegant solution because of the cost and the amount of data required. Specially, the large-scale annotations are hard to obtain for low-resource languages. Additionally, some researchers have found that they could get even better performance on benchmarks when training one model with several languages comparing with training several monolingual models (Lample and Conneau, 2019; Ni *et al.*, 2021). Hence, training PTMs to learn multi-lingual representations rather than monolingual representations may be a better way for IR. In fact, before BERT, some researchers have explored multi-lingual representations for IR (Salton, 1972; Ballesteros and Croft, 1996; Davis and Dunning, 1995). However, these models are usually trained only for one specific task from beginning to end, and cross-lingual knowledge cannot be generalized to other tasks. Hence, for any other multi-lingual tasks, training new models from scratch is still required, needing a large volume of task-specific data. The appearance of PTMs shows that the framework of pre-training with general self-supervised tasks and then fine-tuning on specific downstream tasks is feasible. This motivates researchers to design tasks to pre-train versatile multi-lingual models, such as mBERT (Devlin *et al.*, 2018), XLM (Lample and Conneau, 2019), and Unicoder (Huang *et al.*, 2019). In fact, these multi-lingual PTMs have shown their language transfer abilities over a wide variety of tasks (Wu and Dredze, 2019). For example, Shi *et al.* (2020) built IR re-rankers based on the mBERT for non-English corpus by leveraging the relevance judgments in English, and they found that it significantly improves search quality in (non-English) mono-lingual retrieval. However, most such works on multi-lingual PTMs focus on NLP tasks, and these multi-lingual PTMs are not well designed for cross-lingual tasks in IR.

Knowledge-Enhanced Pre-Training for IR. It is generally accepted that external knowledge, such as knowledge graphs and domain-specific data, can be a good prior to the modeling of statistics. Thus, introducing external knowledge into PTMs to get knowledge-enhanced representations for IR is another research line. The typical form of structured knowledge is knowledge graphs. Based on this, there have been many works trying to enhance PTMs by integrating entity and relation embeddings or their alignments into texts (Zhang *et al.*, 2019c; Sun *et al.*, 2019b; Wang *et al.*, 2021). Compared to structured knowledge, unstructured knowledge, e.g., the data of a specific domain or task, is more abundant but also noisier. Several works (Beltagy *et al.*, 2019; Lee *et al.*, 2020) have attempted to further pre-train the general PTMs on these data to get better domain-specific or task-specific models. However, most of these efforts are not tailored for IR. In the future, how to effectively model this kind of knowledge from the data for IR needs to be further explored. On the other hand, all existing works store knowledge in their model parameters implicitly. How to model external knowledge in a more interpretable way for downstream tasks has not been explored.

8.3 End-to-End IR based on PTMs

Existing IR systems always follow a "index-retrieve-rank" manner and separate three steps during training. However, this pattern suffers from a few drawbacks in practical scenarios and produces sub-optimal performance. Recently, the application of PTMs in the first-stage retrieval makes the joint learning of multi-stages or end-to-end learning possible.

Technically, the index building process in retrieval systems based on the inverted index is hard to be trained jointly with the retrieval stage. However, advances in dense retrieval models resulted in a shift away from the inverted index towards the dense vector-based index makes the joint training possible. In fact, there have been studies (Zhang *et al.*, 2021; Zhan *et al.*, 2021a; Zhan *et al.*, 2021b) to explore the joint training of retrievers and indexes in the field of IR. In this way, the indexing process can benefit from the supervised relevance information directly. In addition to the profit from the joint learning of index and retrieval, there have been works finding that it is beneficial to train retrievers and rankers in a correlated manner. For example, the retriever can be improved by distilling from the ranker with a more capable architecture (Ding *et al.*, 2020; Hofstätter *et al.*, 2020), and the ranker can be improved with training instances generated from the retriever (Gao *et al.*, 2021b; Huang *et al.*, 2020). Based on these observations, Ren *et al.*

(2021) proposed the dynamic listwise distillation to jointly optimize the two modules in order to achieve mutual improvement and contribute to the final ranking performance.

Nevertheless, these works are only a preliminary attempt in this direction. In fact, the two stages, i.e., retrieval and ranking, are usually optimized in different ways so that the joint learning cannot be trivially implemented and many problems remain unresolved. Besides, researchers in this field have not ventured into the end-to-end learning of the whole pipeline (i.e. indexing, retrieval, and ranking). Thus, it is still in its infancy stage to design the joint learning or end-to-end learning schemes, and we believe it would be an interesting and promising direction.

8.4 Next Generation IR System: from Index-centric to Model-centric

Beyond the traditional multi-stage IR systems, the state-of-the-art PTMs with huge model size are capable of encoding more knowledge about the world, and based on this, they are probably able to generate results to information needs directly. Thus, given the significant progress in PTMs, now is an opportune time to envision what possibilities the future might hold in terms of synthesizing and evolving these technologies into the next generation of IR systems.

Recently, Metzler *et al.* (2021) proposed a consolidated model-based approach to building IR systems, referred to as model-based IR. Within the model-based IR systems, indexing is replaced with model training, while retrieval and ranking are replaced with model inference. That is to say, with the model-centric paradigm, the model and the index are one, and everything developed on top of the index previously is now integrated directly into a single, consolidated model. The model itself is built from the corpus, just like indexes are built from the corpus, but the encoded information is expected to be much more complex and able to solve a wider range of IR tasks.

Nevertheless, before the model-based IR system can be realized, there are a number of difficult research and engineering challenges on PTMs must be solved. Here, we only list some of them: (1) existing PTMs are dilettantes — they do not have a true understanding of the world and are prone to hallucinating (Lee *et al.*, 2018a). For example, they are incapable of justifying their utterances by referring to supporting documents in the corpus they are trained over, and it has the potential to result in strange, untrue, or downright offensive outputs. In fact, this is an issue that is prevalent across all neural-based models and needs to be addressed before they are applied in practice; (2) it remains challenging to develop PTMs that are capable of arithmetic, logical, temporal, and geographical reasoning; (3) a desiderata is that the model-based IR system should be interpretable, debuggable and controllable. For example, the model designer should know how to control the behavior of the system, e.g. by modifying training data or tuning hyper-parameters of the model. Besides, we would like the system to be well-behaved for queries it may not have seen before, or even the adversarial inputs.

9

Conclusion

In this paper, we present a comprehensive overview of PTMs in IR, and gain some insights for future development. It includes the background of IR, a detailed description of PTMs applied in different components of IR, and a summary of related resources. Specifically, we describe the concepts of IR in a hierarchical view, and review the major paradigms of each stage. Then we thoroughly survey PTMs applied in different components of IR systems, including the first-stage retrieval component, the re-ranking component, and other components. In addition, we describe works in designing novel PTMs tailored for IR. Finally, we highlight several challenges on this topic and discuss potential research directions in this area. We hope this survey can help researchers who are interested in PTMs in IR, and will motivate new ideas to further explore this promising field.

Acknowledgements

References

Abdul-Jaleel, N., J. Allan, W. B. Croft, F. Diaz, L. Larkey, X. Li, M. D. Smucker, and C. Wade. (2004). “UMass at TREC 2004: Novelty and HARD”. *Computer Science Department Faculty Publication Series*: 189.

Agirre, E., X. Arregi, and A. Otegi. (2010). “Document expansion based on WordNet for robust IR”. In: *Coling 2010: Posters*. 9–17.

Alberti, C., J. Ling, M. Collins, and D. Reitter. (2019). “Fusion of detected objects in text for visual question answering”. *arXiv preprint arXiv:1908.05054*.

Aliannejadi, M., H. Zamani, F. Crestani, and W. B. Croft. (2019). “Asking clarifying questions in open-domain information-seeking conversations”. In: *Proceedings of the 42nd international acm sigir conference on research and development in information retrieval*. 475–484.

Amati, G. and C. Rijsbergen. (2002). “Probabilistic models of information retrieval based on measuring the divergence from randomness”. *ACM Trans. Inf. Syst.*: 357–389.

Amati, G. and C. J. Van Rijsbergen. (2002). “Probabilistic models of information retrieval based on measuring the divergence from randomness”. *ACM Transactions on Information Systems (TOIS)*. 20(4): 357–389.

Antol, S., A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. L. Zitnick, and D. Parikh. (2015). “Vqa: Visual question answering”. In: *Proceedings of the IEEE international conference on computer vision*. 2425–2433.

Bae, S., T. Kim, J. Kim, and S.-g. Lee. (2019). “Summary level training of sentence rewriting for abstractive summarization”. *arXiv preprint arXiv:1909.08752*.

Bai, Y., X. Li, G. Wang, C. Zhang, L. Shang, J. Xu, Z. Wang, F. Wang, and Q. Liu. (2020). “SparTerm: Learning Term-based Sparse Representation for Fast Text Retrieval”. *arXiv preprint arXiv:2010.00768*.

Ballesteros, L. and B. Croft. (1996). “Dictionary methods for cross-lingual information retrieval”. In: *International Conference on Database and Expert Systems Applications*. Springer. 791–801.

Bao, H., L. Dong, F. Wei, W. Wang, N. Yang, X. Liu, Y. Wang, J. Gao, S. Piao, M. Zhou, *et al.* (2020a). “Unilmv2: Pseudo-masked language models for unified language model pre-training”. In: *International Conference on Machine Learning*. PMLR. 642–652.

Bao, S., H. He, F. Wang, and H. Wu. (2020b). “PLATO: Pre-trained Dialogue Generation Model with Discrete Latent Variable”. In: *ACL*.

Baumel, T., M. Eyal, and M. Elhadad. (2018). “Query focused abstractive summarization: Incorporating query relevance, multi-document coverage, and summary length constraints into seq2seq models”. *arXiv preprint arXiv:1801.07704*.

Beltagy, I., K. Lo, and A. Cohan. (2019). “Scibert: A pretrained language model for scientific text”. *arXiv preprint arXiv:1903.10676*.

Bengio, Y., A. Courville, and P. Vincent. (2013). “Representation learning: A review and new perspectives”. *IEEE transactions on pattern analysis and machine intelligence*. 35(8): 1798–1828.

Bi, B., C. Li, C. Wu, M. Yan, W. Wang, S. Huang, F. Huang, and L. Si. (2020a). “Palm: Pre-training an autoencoding&autoregressive language model for context-conditioned generation”. *arXiv preprint arXiv:2004.07159*.

Bi, K., Q. Ai, and W. B. Croft. (2020b). “A Transformer-based Embedding Model for Personalized Product Search”. In: *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1521–1524.

Bi, K., Q. Ai, and W. B. Croft. (2021a). "Asking Clarifying Questions Based on Negative Feedback in Conversational Search". *arXiv preprint arXiv:2107.05760*.

Bi, K., Q. Ai, and W. B. Croft. (2021b). "Learning a Fine-Grained Review-based Transformer Model for Personalized Product Search". In: *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 123–132.

Bi, K., P. Metrikov, C. Li, and B. Byun. (2021c). "Leveraging User Behavior History for Personalized Email Search". In: *Proceedings of the Web Conference 2021*. 2858–2868.

Boualili, L., J. G. Moreno, and M. Boughanem. (2020). "MarkedBERT: Integrating Traditional IR Cues in Pre-trained Language Models for Passage Retrieval". *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*.

Bromley, J., J. W. Bentz, L. Bottou, I. Guyon, Y. LeCun, C. Moore, E. Säckinger, and R. Shah. (1993). "Signature verification using a "siamese" time delay neural network". *International Journal of Pattern Recognition and Artificial Intelligence*. 7(04): 669–688.

Brown, T., B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei. (2020a). "Language Models are Few-Shot Learners". In: *Advances in Neural Information Processing Systems*. 1877–1901.

Brown, T. B., B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei. (2020b). "Language Models are Few-Shot Learners". In: *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*. Ed. by H. Larochelle, M. Ranzato, R. Hadsell, M.-F. Balcan, and H.-T. Lin. URL: https://proceedings.neurips.cc/paper/2020/hash/1457c0d6bfcb4967418bfb8ac142f64a-Abstract.html.

Burges, C., T. Shaked, E. Renshaw, A. Lazier, M. Deeds, N. Hamilton, and G. Hullender. (2005a). "Learning to rank using gradient descent". *Proceedings of the 22nd international conference on Machine learning*.

Burges, C., T. Shaked, E. Renshaw, A. Lazier, M. Deeds, N. Hamilton, and G. Hullender. (2005b). "Learning to rank using gradient descent". In: *Proceedings of the 22nd international conference on Machine learning*. 89–96.

Burges, C., R. Ragno, and Q. Le. (2006). "Learning to rank with nonsmooth cost functions". *Advances in neural information processing systems*. 19: 193–200.

Cai, Y., Y. Fan, J. Guo, F. Sun, R. Zhang, and X. Cheng. (2021a). "Semantic Models for the First-stage Retrieval: A Comprehensive Review". *arXiv preprint arXiv:2103.04831*.

Cai, Y., Y. Fan, J. Guo, F. Sun, R. Zhang, and X. Cheng. (2021b). "Semantic Models for the First-stage Retrieval: A Comprehensive Review". *CoRR*. abs/2103.04831. arXiv: 2103.04831. URL: https://arxiv.org/abs/2103.04831.

Cao, J., Z. Gan, Y. Cheng, L. Yu, Y.-C. Chen, and J. Liu. (2020). "Behind the scene: Revealing the secrets of pre-trained vision-and-language models". In: *European Conference on Computer Vision*. Springer. 565–580.

Carbonell, J. and J. Goldstein. (1998). “The use of MMR, diversity-based reranking for reordering documents and producing summaries”. In: *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*. 335–336.

Chang, W.-C., F. X. Yu, Y.-W. Chang, Y. Yang, and S. Kumar. (2019). “Pre-training Tasks for Embedding-based Large-scale Retrieval”. In: *ICLR*.

Chen, J., J. Mao, Y. Liu, Z. Ye, W. Ma, C. Wang, M. Zhang, and S. Ma. (2021a). “A Hybrid Framework for Session Context Modeling”. *ACM Transactions on Information Systems (TOIS)*. 39(3): 1–35.

Chen, R.-C. and C.-J. Lee. (2020). “Incorporating Context Structures for Query Generation”. In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 3105–3110.

Chen, W.-F., S. Syed, B. Stein, M. Hagen, and M. Potthast. (2020a). “Abstractive Snippet Generation”. In: *Proceedings of The Web Conference 2020*. 1309–1319.

Chen, X., X. Xie, N. Zhang, J. Yan, S. Deng, C. Tan, F. Huang, L. Si, and H. Chen. (2021b). “Adaprompt: Adaptive prompt-based finetuning for relation extraction”. *arXiv preprint arXiv:2104.07650*.

Chen, X., B. He, K. Hui, L. Sun, and Y. Sun. (2021c). “Simplified TinyBERT: Knowledge Distillation for Document Retrieval”. In: *ECIR*.

Chen, Z., X. Fan, and Y. Ling. (2020b). “Pre-training for query rewriting in a spoken language understanding system”. In: *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 7969–7973.

Chiang, W.-T. M., M. Hagenbuchner, and A. C. Tsoi. (2005). “The WT10G Dataset and the Evolution of the Web”. In: *Special interest tracks and posters of the 14th international conference on World Wide Web*. 938–939.

Choi, J., E. Jung, J. Suh, and W. Rhee. (2021). “Improving Bi-encoder Document Ranking Models with Two Rankers and Multi-teacher Distillation”. *arXiv preprint arXiv:2103.06523*.

Chuang, Y.-S., C.-L. Liu, H.-Y. Lee, and L.-s. Lee. (2019). "Speechbert: An audio-and-text jointly learned language model for end-to-end spoken question answering". *arXiv preprint arXiv:1910.11559*.

Conneau, A. and D. Kiela. (2018). "Senteval: An evaluation toolkit for universal sentence representations". *arXiv preprint arXiv:1803.05449*.

Crammer, K., Y. Singer, *et al.* (2001). "Pranking with ranking." In: *Nips*. Vol. 1. 641–647.

Craswell, N., B. Mitra, E. Yilmaz, D. Campos, and E. M. Voorhees. (2020). "Overview of the trec 2019 deep learning track". *arXiv preprint arXiv:2003.07820*.

Craswell, N., B. Mitra, E. Yilmaz, D. F. Campos, and J. J. Lin. (2021). "MS MARCO: Benchmarking Ranking Models in the Large-Data Regime". *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*.

Croft, W. B., D. Metzler, and T. Strohman. (2009). "Search Engines - Information Retrieval in Practice". In:

Dai, Z. and J. Callan. (2019a). "Context-aware sentence/passage term importance estimation for first stage retrieval". *arXiv preprint arXiv:1910.10687*.

Dai, Z. and J. Callan. (2019b). "Deeper Text Understanding for IR with Contextual Neural Language Modeling". *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*.

Dai, Z. and J. Callan. (2020a). "Context-Aware Document Term Weighting for Ad-Hoc Search". In: *Proceedings of The Web Conference 2020*. 1897–1907.

Dai, Z. and J. Callan. (2020b). "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*. 1533–1536.

Dang, H. T. (2005). "Overview of DUC 2005". In: *Proceedings of the document understanding conference*. Vol. 2005. 1–12.

Davis, M. W. and T. Dunning. (1995). "Query Translation Using Evolutionary Programming for Multi-lingual Information Retrieval." In: *Evolutionary Programming*. 175–185.

Dehghani, M., S. Rothe, E. Alfonseca, and P. Fleury. (2017). "Learning to attend, copy, and generate for session-based query suggestion". In: *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. 1747–1756.

Devlin, J., M.-W. Chang, K. Lee, and K. Toutanova. (2018). "Bert: Pre-training of deep bidirectional transformers for language understanding". *arXiv preprint arXiv:1810.04805*.

Dietz, L., M. Verma, F. Radlinski, and N. Craswell. (2017). "TREC Complex Answer Retrieval Overview." In: *TREC*.

Ding, Y., Y. Qu, J. Liu, K. Liu, R. Ren, X. Zhao, D. Dong, H. Wu, and H. Wang. (2020). "RocketQA: An Optimized Training Approach to Dense Passage Retrieval for Open-Domain Question Answering". *arXiv preprint arXiv:2010.08191*.

Dong, L., N. Yang, W. Wang, F. Wei, X. Liu, Y. Wang, J. Gao, M. Zhou, and H. Hon. (2019). "Unified Language Model Pre-training for Natural Language Understanding and Generation". *ArXiv*. abs/1905.03197.

Dou, Z.-Y., P. Liu, H. Hayashi, Z. Jiang, and G. Neubig. (2020). "Gsum: A general framework for guided neural abstractive summarization". *arXiv preprint arXiv:2010.08014*.

Dubey, S. R. (2020). "A Decade Survey of Content Based Image Retrieval using Deep Learning". *CoRR*. abs/2012.00641. arXiv: 2012.00641. URL: https://arxiv.org/abs/2012.00641.

Efron, M., P. Organisciak, and K. 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.

Erkan, G. and D. R. Radev. (2004). "Lexrank: Graph-based lexical centrality as salience in text summarization". *Journal of artificial intelligence research*. 22: 457–479.

Fedus, W., B. Zoph, and N. Shazeer. (2021). "Switch Transformers: Scaling to Trillion Parameter Models with Simple and Efficient Sparsity". *CoRR*. abs/2101.03961. arXiv: 2101.03961. URL: https://arxiv.org/abs/2101.03961.

Fei, H., T. Yu, and P. Li. (2021). "Cross-lingual Cross-modal Pretraining for Multimodal Retrieval". In: *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*. Ed. by K. Toutanova, A. Rumshisky, L. Zettlemoyer, D. Hakkani-Tür, I. Beltagy, S. Bethard, R. Cotterell, T. Chakraborty, and Y. Zhou. Association for Computational Linguistics. 3644–3650. DOI: 10.18653/v1/2021.naacl-main.285. URL: https://doi.org/10.18653/v1/2021.naacl-main.285.

Feigenblat, G., H. Roitman, O. Boni, and D. Konopnicki. (2017). "Unsupervised query-focused multi-document summarization using the cross entropy method". In: *Proceedings of the 40th International ACM SIGIR Conference on research and development in information retrieval*. 961–964.

Fisher, S. and B. Roark. (2006). "Query-focused summarization by supervised sentence ranking and skewed word distributions". In: *Proceedings of the Document Understanding Conference, DUC-2006, New York, USA*. Citeseer.

Formal, T., B. Piwowarski, and S. Clinchant. (2021). "SPLADE: Sparse Lexical and Expansion Model for First Stage Ranking". In: *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2288–2292.

Gao, L. and J. Callan. (2021a). "Condenser: a Pre-training Architecture for Dense Retrieval". In: *EMNLP*.

Gao, L. and J. Callan. (2021b). "Is Your Language Model Ready for Dense Representation Fine-tuning?" *arXiv preprint arXiv:2104.08253*.

Gao, L. and J. Callan. (2021c). "Unsupervised Corpus Aware Language Model Pre-training for Dense Passage Retrieval". *arXiv preprint arXiv:2108.05540*.

Gao, L., Z. Dai, and J. Callan. (2020a). "Understanding BERT Rankers Under Distillation". *Proceedings of the 2020 ACM SIGIR on International Conference on Theory of Information Retrieval*.

Gao, L., Z. Dai, and J. Callan. (2021a). "COIL: Revisit Exact Lexical Match in Information Retrieval with Contextualized Inverted List". *arXiv preprint arXiv:2104.07186*.

Gao, L., Z. Dai, and J. Callan. (2021b). “Rethink training of BERT rerankers in multi-stage retrieval pipeline”. *arXiv preprint arXiv:2101.08751*.

Gao, L., Z. Dai, Z. Fan, and J. Callan. (2020b). “Complementing Lexical Retrieval with Semantic Residual Embedding”. *arXiv preprint arXiv:2004.13969*.

Gao, T., A. Fisch, and D. Chen. (2020c). “Making pre-trained language models better few-shot learners”. *arXiv preprint arXiv:2012.15723*.

Glass, M. R., A. Gliozzo, R. Chakravarti, A. Ferritto, L. Pan, G. P. S. Bhargav, D. Garg, and A. Sil. (2020). “Span Selection Pre-training for Question Answering”. In: *ACL*.

Grover, A. and J. Leskovec. (2016). “node2vec: Scalable feature learning for networks”. In: *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*. 855–864.

Guo, J., Y. Fan, Q. Ai, and W. Croft. (2016a). “A Deep Relevance Matching Model for Ad-hoc Retrieval”. *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*.

Guo, J., Y. Fan, Q. Ai, and W. B. Croft. (2016b). “A deep relevance matching model for ad-hoc retrieval”. In: *Proceedings of the 25th ACM international on conference on information and knowledge management*. 55–64.

Guo, J., Y. Fan, L. Pang, L. Yang, Q. Ai, H. Zamani, C. Wu, W. B. Croft, and X. Cheng. (2020a). “A deep look into neural ranking models for information retrieval”. *Information Processing & Management*. 57(6): 102067.

Guo, J., Y. Fan, L. Pang, L. Yang, Q. Ai, H. Zamani, C. Wu, W. B. Croft, and X. Cheng. (2020b). “A Deep Look into neural ranking models for information retrieval”. *Inf. Process. Manag.* 57(6): 102067. DOI: 10.1016/j.ipm.2019.102067. URL: https://doi.org/10.1016/j.ipm.2019.102067.

Guo, J., Y. Fan, L. Pang, L. Yang, Q. Ai, H. Zamani, C. Wu, W. Croft, and X. Cheng. (2020c). “A Deep Look into Neural Ranking Models for Information Retrieval”. *ArXiv*. abs/1903.06902.

Guu, K., K. Lee, Z. Tung, P. Pasupat, and M.-W. Chang. (2020). "Realm: Retrieval-augmented language model pre-training". In: *Proceedings of the 37th International Conference on Machine Learning.*

Han, X., W. Zhao, N. Ding, Z. Liu, and M. Sun. (2021a). "PTR: Prompt Tuning with Rules for Text Classification". *arXiv preprint arXiv:2105.11259.*

Han, Y., G. Huang, S. Song, L. Yang, H. Wang, and Y. Wang. (2021b). "Dynamic Neural Networks: A Survey". *ArXiv.* abs/2102.04906.

Hashemi, H., H. Zamani, and W. B. Croft. (2020). "Guided transformer: Leveraging multiple external sources for representation learning in conversational search". In: *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval.* 1131–1140.

Herbrich, R., T. Graepel, and K. Obermayer. (1999). "Support vector learning for ordinal regression".

Hinton, G. E., O. Vinyals, and J. Dean. (2015). "Distilling the Knowledge in a Neural Network". *ArXiv.* abs/1503.02531.

Hofstätter, S., S. Althammer, M. Schröder, M. Sertkan, and A. Hanbury. (2020). "Improving efficient neural ranking models with cross-architecture knowledge distillation". *arXiv preprint arXiv:2010.02666.*

Hofstätter, S., S.-C. Lin, J.-H. Yang, J. Lin, and A. Hanbury. (2021a). "Efficiently Teaching an Effective Dense Retriever with Balanced Topic Aware Sampling". *arXiv preprint arXiv:2104.06967.*

Hofstätter, S., B. Mitra, H. Zamani, N. Craswell, and A. Hanbury. (2021b). "Intra-Document Cascading: Learning to Select Passages for Neural Document Ranking". *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval.*

Howard, J. and S. Ruder. (2018). "Universal Language Model Fine-tuning for Text Classification". In: *ACL.*

Huang, H., Y. Liang, N. Duan, M. Gong, L. Shou, D. Jiang, and M. Zhou. (2019). "Unicoder: A universal language encoder by pre-training with multiple cross-lingual tasks". *arXiv preprint arXiv:1909.00964.*

Huang, J.-T., A. Sharma, S. Sun, L. Xia, D. Zhang, P. Pronin, J. Padmanabhan, G. Ottaviano, and L. Yang. (2020). "Embedding-based retrieval in facebook search". In: *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2553–2561.

Huang, P.-S., X. He, J. Gao, L. Deng, A. Acero, and L. Heck. (2013). "Learning deep structured semantic models for web search using clickthrough data". In: *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*. 2333–2338.

Huang, Y., Z. Yu, J. Guo, Y. Xiang, and Y. Xian. (2021). "Element Graph-Augmented Abstractive Summarization for Legal Public Opinion News with Graph Transformer". *Neurocomputing*.

Huo, Y., M. Zhang, G. Liu, H. Lu, Y. Gao, G. Yang, J. Wen, H. Zhang, B. Xu, W. Zheng, *et al.* (2021). "WenLan: Bridging vision and language by large-scale multi-modal pre-training". *arXiv preprint arXiv:2103.06561*.

Jang, K.-R., J. Kang, G. Hong, S.-H. Myaeng, J. Park, T. Yoon, and H. Seo. (2021). "UHD-BERT: Bucketed Ultra-High Dimensional Sparse Representations for Full Ranking". *arXiv preprint arXiv:2104.07198*.

Jegou, H., M. Douze, and C. Schmid. (2010). "Product quantization for nearest neighbor search". *IEEE transactions on pattern analysis and machine intelligence*. 33(1): 117–128.

Jiang, J.-Y. and W. Wang. (2018). "RIN: Reformulation inference network for context-aware query suggestion". In: *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. 197–206.

Jiang, Z., F. F. Xu, J. Araki, and G. Neubig. (2020). "How can we know what language models know?" *Transactions of the Association for Computational Linguistics*. 8: 423–438.

Jiao, X., Y. Yin, L. Shang, X. Jiang, X. Chen, L. Li, F. Wang, and Q. Liu. (2020). "TinyBERT: Distilling BERT for Natural Language Understanding". In: *FINDINGS*.

Joachims, T. (2002). "Optimizing search engines using clickthrough data". *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*.

Johnson, J., A. Karpathy, and L. Fei-Fei. (2016). "Densecap: Fully convolutional localization networks for dense captioning". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 4565–4574.

Joshi, M., E. Choi, D. S. Weld, and L. Zettlemoyer. (2017). "TriviaQA: A Large Scale Distantly Supervised Challenge Dataset for Reading Comprehension". *CoRR*. abs/1705.03551. arXiv: 1705.03551. URL: http://arxiv.org/abs/1705.03551.

Karpukhin, V., B. Oğuz, S. Min, L. Wu, S. Edunov, D. Chen, and W.-t. Yih. (2020). "Dense Passage Retrieval for Open-Domain Question Answering". *arXiv preprint arXiv:2004.04906*.

Ke, P., H. Ji, S. Liu, X. Zhu, and M. Huang. (2020a). "Sentilare: Linguistic knowledge enhanced language representation for sentiment analysis". In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 6975–6988.

Ke, P., H. Ji, S. Liu, X. Zhu, and M. Huang. (2020b). "SentiLARE: Sentiment-Aware Language Representation Learning with Linguistic Knowledge". In: *EMNLP*.

Khattab, O. and M. Zaharia. (2020). "Colbert: Efficient and effective passage search via contextualized late interaction over bert". In: *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*. 39–48.

Kulkarni, S., S. Chammas, W. Zhu, F. Sha, and E. Ie. (2020). "Aquamuse: Automatically generating datasets for query-based multi-document summarization". *arXiv preprint arXiv:2010.12694*.

Kumar, V., V. Raunak, and J. Callan. (2020). "Ranking Clarification Questions via Natural Language Inference". In: *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 2093–2096.

Kurland, O. and L. 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.

Kuzi, S., M. Zhang, C. Li, M. Bendersky, and M. Najork. (2020). "Leveraging Semantic and Lexical Matching to Improve the Recall of Document Retrieval Systems: A Hybrid Approach". *arXiv preprint arXiv:2010.01195*.

Kwiatkowski, T., J. Palomaki, O. Redfield, M. Collins, A. Parikh, C. Alberti, D. Epstein, I. Polosukhin, J. Devlin, K. Lee, *et al.* (2019). "Natural questions: a benchmark for question answering research". *Transactions of the Association for Computational Linguistics*. 7: 453–466.

Lafferty, J. and C. Zhai. (2001). "Document language models, query models, and risk minimization for information retrieval". In: *SIGIR '01*.

Lafferty, J. and C. Zhai. (2003). "Probabilistic Relevance Models Based on Document and Query Generation". In:

Lample, G. and A. Conneau. (2019). "Cross-lingual language model pretraining". *arXiv preprint arXiv:1901.07291*.

Laskar, M. T. R., E. Hoque, and J. Huang. (2020a). "Query focused abstractive summarization via incorporating query relevance and transfer learning with transformer models". In: *Canadian Conference on Artificial Intelligence*. Springer. 342–348.

Laskar, M. T. R., E. Hoque, and J. X. Huang. (2020b). "WSL-DS: Weakly supervised learning with distant supervision for query focused multi-document abstractive summarization". *arXiv preprint arXiv:2011.01421*.

Lee, J., M. Seo, H. Hajishirzi, and J. Kang. (2019a). "Contextualized sparse representations for real-time open-domain question answering". *arXiv preprint arXiv:1911.02896*.

Lee, J., W. Yoon, S. Kim, D. Kim, S. Kim, C. H. So, and J. Kang. (2020). "BioBERT: a pre-trained biomedical language representation model for biomedical text mining". *Bioinformatics*. 36(4): 1234–1240.

Lee, K., O. Firat, A. Agarwal, C. Fannjiang, and D. Sussillo. (2018a). "Hallucinations in neural machine translation".

Lee, K., M.-W. Chang, and K. Toutanova. (2019b). "Latent Retrieval for Weakly Supervised Open Domain Question Answering". In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Stroudsburg, PA, USA: ACL. 6086–6096.

Lee, K.-H., X. Chen, G. Hua, H. Hu, and X. He. (2018b). “Stacked cross attention for image-text matching”. In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 201–216.

Lester, B., R. Al-Rfou, and N. Constant. (2021). “The power of scale for parameter-efficient prompt tuning”. *arXiv preprint arXiv:2104.08691*.

Lewis, M., Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer. (2019). “Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension”. *arXiv preprint arXiv:1910.13461*.

Lewis, P., E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel, *et al.* (2020). “Retrieval-augmented generation for knowledge-intensive nlp tasks”. *arXiv preprint arXiv:2005.11401*.

Li, C., A. Yates, S. MacAvaney, B. He, and Y. Sun. (2020a). “PARADE: Passage Representation Aggregation for Document Reranking”. *ArXiv*. abs/2008.09093.

Li, H. (2011). *Learning to Rank for Information Retrieval and Natural Language Processing*. Morgan & Claypool Publishers. ISBN: 1608457079, 9781608457076.

Li, H. (2014). “Learning to rank for information retrieval and natural language processing”. *Synthesis lectures on human language technologies*. 7(3): 1–121.

Li, H. and J. Xu. (2014). “Semantic Matching in Search”. In: *SMIR@SIGIR*.

Li, L. H., M. Yatskar, D. Yin, C.-J. Hsieh, and K.-W. Chang. (2019). “Visualbert: A simple and performant baseline for vision and language”. *arXiv preprint arXiv:1908.03557*.

Li, P., C. J. C. Burges, and Q. Wu. (2007). “McRank: Learning to Rank Using Multiple Classification and Gradient Boosting”. In: *NIPS*.

Li, W., C. Gao, G. Niu, X. Xiao, H. Liu, J. Liu, H. Wu, and H. Wang. (2020b). “Unimo: Towards unified-modal understanding and generation via cross-modal contrastive learning”. *arXiv preprint arXiv:2012.15409*.

Li, X., X. Yin, C. Li, P. Zhang, X. Hu, L. Zhang, L. Wang, H. Hu, L. Dong, F. Wei, *et al.* (2020c). “Oscar: Object-semantics aligned pre-training for vision-language tasks”. In: *European Conference on Computer Vision*. Springer. 121–137.

Liang, D., P. Xu, S. Shakeri, C. N. d. Santos, R. Nallapati, Z. Huang, and B. Xiang. (2020). "Embedding-based Zero-shot Retrieval through Query Generation". *arXiv preprint arXiv:2009.10270*.

Lim, Y., D. Seo, and Y. Jung. (2020). "Fine-tuning BERT Models for Keyphrase Extraction in Scientific Articles". *JOURNAL OF ADVANCED INFORMATION TECHNOLOGY AND CONVERGENCE*. 10(1): 45–56.

Lin, J. and X. Ma. (2021). "A Few Brief Notes on DeepImpact, COIL, and a Conceptual Framework for Information Retrieval Techniques". *arXiv preprint arXiv:2106.14807*.

Lin, J. J., R. Nogueira, and A. Yates. (2021). "Pretrained Transformers for Text Ranking: BERT and Beyond". *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*.

Lin, S.-C., J.-H. Yang, and J. Lin. (2020a). "Distilling Dense Representations for Ranking using Tightly-Coupled Teachers". *arXiv preprint arXiv:2010.11386*.

Lin, S.-C., J.-H. Yang, R. Nogueira, M.-F. Tsai, C.-J. Wang, and J. Lin. (2020b). "Query reformulation using query history for passage retrieval in conversational search". *arXiv preprint arXiv:2005.02230*.

Liu, B., H. Zamani, X. Lu, and J. Culpepper. (2021a). "Generalizing Discriminative Retrieval Models using Generative Tasks". *Proceedings of the Web Conference 2021*.

Liu, H., M. Chen, Y. Wu, X. He, and B. Zhou. (2021b). "Conversational Query Rewriting with Self-Supervised Learning". In: *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 7628–7632.

Liu, R., Z. Lin, and W. Wang. (2020). "Keyphrase prediction with pre-trained language model". *arXiv preprint arXiv:2004.10462*.

Liu, T.-Y. (2010). "Learning to rank for information retrieval". *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*.

Liu, X., Y. Zheng, Z. Du, M. Ding, Y. Qian, Z. Yang, and J. Tang. (2021c). "GPT Understands, Too". *arXiv preprint arXiv:2103.10385*.

Liu, X. and W. B. 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.

Liu, X. and W. B. Croft. (2005). "Statistical Language Modeling for Information Retrieval". In: *IR, UMASS.*

Liu, Y. (2019). "Fine-tune BERT for extractive summarization". *arXiv preprint arXiv:1903.10318.*

Liu, Y. and M. Lapata. (2019). "Text summarization with pretrained encoders". *arXiv preprint arXiv:1908.08345.*

Liu, Y., G. Huang, J. Liu, W. Lu, S. Cheng, Y. Li, D. Shi, S. Wang, Z. Cheng, and D. Yin. (2021d). "Pre-trained Language Model for Web-scale Retrieval in Baidu Search". *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining.*

Liu, Y., W. Lu, S. Cheng, D. Shi, S. Wang, Z. Cheng, and D. Yin. (2021e). "Pre-trained Language Model for Web-scale Retrieval in Baidu Search". *arXiv preprint arXiv:2106.03373.*

Liu, Y., M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov. (2019). "Roberta: A robustly optimized bert pretraining approach". *arXiv preprint arXiv:1907.11692.*

Lotze, T., S. Klut, M. Aliannejadi, and E. Kanoulas. (2021). "Ranking Clarifying Questions Based on Predicted User Engagement". *arXiv preprint arXiv:2103.06192.*

Lu, J., D. Batra, D. Parikh, and S. Lee. (2019). "Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks". *arXiv preprint arXiv:1908.02265.*

Lu, S., C. Xiong, D. He, G. Ke, W. Malik, Z. Dou, P. Bennett, T. Liu, and A. Overwijk. (2021). "Less is More: Pre-training a Strong Siamese Encoder Using a Weak Decoder". *arXiv preprint arXiv:2102.09206.*

Luan, Y., J. Eisenstein, K. Toutanova, and M. Collins. (2020). "Sparse, Dense, and Attentional Representations for Text Retrieval". *arXiv preprint arXiv:2005.00181.*

Luo, C., Y. Zheng, Y. Liu, X. Wang, J. Xu, M. Zhang, and S. Ma. (2017). "Sogout-16: A new web corpus to embrace ir research". In: *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval.* 1233–1236.

Ma, J., I. Korotkov, Y. Yang, K. Hall, and R. McDonald. (2021a). "Zero-shot Neural Passage Retrieval via Domain-targeted Synthetic Question Generation". In: *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*. 1075–1088.

Ma, X., J. Guo, R. Zhang, Y. Fan, X. Ji, and X. Cheng. (2021b). "PROP: Pre-training with Representative Words Prediction for Ad-hoc Retrieval". *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*.

Ma, X., J. Guo, R. Zhang, Y. Fan, Y. Li, and X. Cheng. (2021c). "B-PROP: Bootstrapped Pre-training with Representative Words Prediction for Ad-hoc Retrieval". *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*.

Ma, Z., Z. Dou, W. Xu, X. Zhang, H. Jiang, Z. Cao, and J.-R. Wen. (2021d). "Pre-training for Ad-hoc Retrieval: Hyperlink is Also You Need". In: *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. 1212–1221.

MacAvaney, S., F. M. Nardini, R. Perego, N. Tonellotto, N. Goharian, and O. Frieder. (2020). "Efficient Document Re-Ranking for Transformers by Precomputing Term Representations". *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*.

MacAvaney, S., A. Yates, A. Cohan, and N. Goharian. (2019a). "CEDR: Contextualized Embeddings for Document Ranking". *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*.

MacAvaney, S., A. Yates, A. Cohan, and N. Goharian. (2019b). "CEDR: Contextualized embeddings for document ranking". In: *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1101–1104.

Mallia, A., O. Khattab, N. Tonellotto, and T. Suel. (2021). "Learning Passage Impacts for Inverted Indexes". *arXiv preprint arXiv:2104.12016*.

Manning, C. D., P. Raghavan, and H. Schütze. (2005). "Introduction to information retrieval". In:

Mao, Y., P. He, X. Liu, Y. Shen, J. Gao, J. Han, and W. Chen. (2020). "Generation-Augmented Retrieval for Open-domain Question Answering". *arXiv preprint arXiv:2009.08553*.

Mass, Y., B. Carmeli, H. Roitman, and D. Konopnicki. (2020). "Unsupervised FAQ retrieval with question generation and BERT". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. 807–812.

McCreery, C. H., N. Katariya, A. Kannan, M. Chablani, and X. Amatriain. (2020). "Effective transfer learning for identifying similar questions: matching user questions to COVID-19 FAQs". In: *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 3458–3465.

Mei, Q., J. Guo, and D. Radev. (2010). "Divrank: the interplay of prestige and diversity in information networks". In: *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*. 1009–1018.

Metzler, D., Y. Tay, D. Bahri, and M. Najork. (2021). "Rethinking search: making domain experts out of dilettantes". In: *ACM SIGIR Forum*. Vol. 55. No. 1. ACM New York, NY, USA. 1–27.

Mikolov, T., K. Chen, G. Corrado, and J. Dean. (2013). "Efficient estimation of word representations in vector space". *arXiv preprint arXiv:1301.3781*.

Mitra, B., F. Diaz, and N. Craswell. (2017). "Learning to match using local and distributed representations of text for web search". In: *Proceedings of the 26th International Conference on World Wide Web*. 1291–1299.

Mitra, R., M. Gupta, and S. Dandapat. (2020). "Transformer Models for Recommending Related Questions in Web Search". In: *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 2153–2156.

Mustar, A., S. Lamprier, and B. Piwowarski. (2020). "Using BERT and BART for Query Suggestion." In: *CIRCLE*.

Naseri, S., J. Dalton, A. Yates, and J. Allan. (2021). "CEQE: Contextualized Embeddings for Query Expansion". *arXiv preprint arXiv:2103.05256*.

Nema, P., M. Khapra, A. Laha, and B. Ravindran. (2017). "Diversity driven attention model for query-based abstractive summarization". *arXiv preprint arXiv:1704.08300*.

Nguyen, T., M. Rosenberg, X. Song, J. Gao, S. Tiwary, R. Majumder, and L. Deng. (2016). "MS MARCO: A human generated machine reading comprehension dataset". In: *CoCo@ NIPS*.

Ni, M., H. Huang, L. Su, E. Cui, T. Bharti, L. Wang, D. Zhang, and N. Duan. (2021). "M3p: Learning universal representations via multitask multilingual multimodal pre-training". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 3977–3986.

Nogueira, R. and K. Cho. (2019). "Passage Re-ranking with BERT". *ArXiv*. abs/1901.04085.

Nogueira, R., Z. Jiang, R. Pradeep, and J. J. Lin. (2020). "Document Ranking with a Pretrained Sequence-to-Sequence Model". In: *FINDINGS*.

Nogueira, R., J. Lin, and A. Epistemic. (2019a). "From doc2query to docTTTTTquery". *Technical report*.

Nogueira, R., W. Yang, K. Cho, and J. J. Lin. (2019b). "Multi-Stage Document Ranking with BERT". *ArXiv*. abs/1910.14424.

Onal, K. D., Y. Zhang, I. S. Altingovde, M. M. Rahman, P. Karagoz, A. Braylan, B. Dang, H.-L. Chang, H. Kim, Q. McNamara, *et al.* (2018a). "Neural information retrieval: At the end of the early years". *Information Retrieval Journal*. 21(2): 111–182.

Onal, K. D., Y. Zhang, I. S. Altingovde, M. M. Rahman, P. Karagoz, A. Braylan, B. Dang, H.-L. Chang, H. Kim, Q. McNamara, A. Angert, E. Banner, V. Khetan, T. McDonnell, A. T. Nguyen, D. Xu, B. C. Wallace, M. de Rijke, and M. Lease. (2018b). "Neural information retrieval: at the end of the early years". *Inf. Retr. J.* 21(2-3): 111–182. DOI: 10.1007/s10791-017-9321-y. URL: https://doi.org/10.1007/s10791-017-9321-y.

Padaki, R., Z. Dai, and J. Callan. (2020). "Rethinking query expansion for bert reranking". *Advances in Information Retrieval*. 12036: 297.

Park, S. and C. Caragea. (2020). “Scientific keyphrase identification and classification by pre-trained language models intermediate task transfer learning”. In: *Proceedings of the 28th International Conference on Computational Linguistics*. 5409–5419.

Pennington, J., R. Socher, and C. D. Manning. (2014). “Glove: Global vectors for word representation”. In: *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. 1532–1543.

Peters, M. E., M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer. (2018a). “Deep contextualized word representations”. *arXiv preprint arXiv:1802.05365*.

Peters, M. E., M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer. (2018b). “Deep Contextualized Word Representations”. In: *NAACL*.

Petroni, F., T. Rocktäschel, P. Lewis, A. Bakhtin, Y. Wu, A. H. Miller, and S. Riedel. (2019). “Language models as knowledge bases?” *arXiv preprint arXiv:1909.01066*.

Ponte, J. M. and W. B. Croft. (1998). “A Language Modeling Approach to Information Retrieval”. In: *Proceedings of the 21th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. New York, NY, USA: ACM. 275–281.

Pradeep, R., R. Nogueira, and J. J. Lin. (2021). “The Expando-Mono-Duo Design Pattern for Text Ranking with Pretrained Sequence-to-Sequence Models”. *ArXiv*. abs/2101.05667.

Prakash, P., J. Killingback, and H. Zamani. (2021). “Learning Robust Dense Retrieval Models from Incomplete Relevance Labels”. In: *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1728–1732.

Puri, R. and B. Catanzaro. (2019). “Zero-shot text classification with generative language models”. *arXiv preprint arXiv:1912.10165*.

Qiao, Y., C. Xiong, Z. Liu, and Z. Liu. (2019). “Understanding the Behaviors of BERT in Ranking”. *ArXiv*. abs/1904.07531.

Qiu, X., T. Sun, Y. Xu, Y. Shao, N. Dai, and X. Huang. (2020). “Pre-trained Models for Natural Language Processing: A Survey”. *ArXiv*. abs/2003.08271.

Radford, A., J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever. (2019). "Language Models are Unsupervised Multitask Learners". In:

Raffel, C., N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu. (2019). "Exploring the limits of transfer learning with a unified text-to-text transformer". *arXiv preprint arXiv:1910.10683*.

Raffel, C., N. M. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu. (2020). "Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer". *ArXiv*. abs/1910.10683.

Reddy, R. G., V. Yadav, M. A. Sultan, M. Franz, V. Castelli, H. Ji, and A. Sil. (2021). "Towards Robust Neural Retrieval Models with Synthetic Pre-Training". *arXiv preprint arXiv:2104.07800*.

Ren, R., S. Lv, Y. Qu, J. Liu, W. X. Zhao, Q. She, H. Wu, H. Wang, and J.-R. Wen. "PAIR: Leveraging Passage-Centric Similarity Relation for Improving Dense Passage Retrieval".

Ren, R., Y. Qu, J. Liu, W. X. Zhao, Q. She, H. Wu, H. Wang, and J.-R. Wen. (2021). "RocketQAv2: A Joint Training Method for Dense Passage Retrieval and Passage Re-ranking". *arXiv preprint arXiv:2110.07367*.

Robertson, S. and K. Jones. (1976a). "Relevance weighting of search terms". *J. Am. Soc. Inf. Sci.* 27: 129–146.

Robertson, S. and S. Walker. (1994). "Some simple effective approximations to the 2-Poisson model for probabilistic weighted retrieval". In: *SIGIR '94*.

Robertson, S. and H. Zaragoza. (2009). *The probabilistic relevance framework: BM25 and beyond*. Now Publishers Inc.

Robertson, S. E. and K. S. Jones. (1976b). "Relevance weighting of search terms". *Journal of the American Society for Information science*. 27(3): 129–146.

Robertson, S. E., S. Walker, S. Jones, M. M. Hancock-Beaulieu, M. Gatford, *et al.* (1995). "Okapi at TREC-3". *Nist Special Publication Sp*. 109: 109.

Roitman, H., G. Feigenblat, D. Cohen, O. Boni, and D. Konopnicki. (2020). "Unsupervised dual-cascade learning with pseudo-feedback distillation for query-focused extractive summarization". In: *Proceedings of The Web Conference 2020*. 2577–2584.

Rosset, C., C. Xiong, X. Song, D. Campos, N. Craswell, S. Tiwary, and P. Bennett. (2020). "Leading conversational search by suggesting useful questions". In: *Proceedings of The Web Conference 2020*. 1160–1170.

Sachan, D. S., S. Reddy, W. Hamilton, C. Dyer, and D. Yogatama. (2021). "End-to-End Training of Multi-Document Reader and Retriever for Open-Domain Question Answering". *arXiv preprint arXiv:2106.05346*.

Sahrawat, D., D. Mahata, H. Zhang, M. Kulkarni, A. Sharma, R. Gosangi, A. Stent, Y. Kumar, R. R. Shah, and R. Zimmermann. (2020). "Keyphrase extraction as sequence labeling using contextualized embeddings". *Advances in Information Retrieval*. 12036: 328.

Saito, I., K. Nishida, K. Nishida, and J. Tomita. (2020). "Abstractive summarization with combination of pre-trained sequence-to-sequence and saliency models". *arXiv preprint arXiv:2003.13028*.

Sakata, W., T. Shibata, R. Tanaka, and S. Kurohashi. (2019). "FAQ retrieval using query-question similarity and BERT-based query-answer relevance". In: *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1113–1116.

Salton, G., A. Wong, and C. S. Yang. (1975). "A Vector Space Model for Automatic Indexing". *Commun. ACM*. 18(11): 613–620. ISSN: 0001-0782.

Salton, G. (1972). "Experiments in multi-lingual information retrieval". *Tech. rep.* Cornell University.

Sanh, V., L. Debut, J. Chaumond, and T. Wolf. (2019). "DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter". *ArXiv*. abs/1910.01108.

Santos, C. D., X. Ma, R. Nallapati, Z. Huang, and B. Xiang. (2020). "Beyond [CLS] through Ranking by Generation". In: *EMNLP*.

Savery, M., A. B. Abacha, S. Gayen, and D. Demner-Fushman. (2020). "Question-driven summarization of answers to consumer health questions". *Scientific Data*. 7(1): 1–9.

Schick, T. and H. Schütze. (2020). "Exploiting cloze questions for few shot text classification and natural language inference". *arXiv preprint arXiv:2001.07676*.

Seo, M., J. Lee, T. Kwiatkowski, A. P. Parikh, A. Farhadi, and H. Hajishirzi. (2019). "Real-time open-domain question answering with dense-sparse phrase index". *arXiv preprint arXiv:1906.05807*.

Shen, Y., X. He, J. Gao, L. Deng, and G. Mesnil. (2014). "A latent semantic model with convolutional-pooling structure for information retrieval". In: *Proceedings of the 23rd ACM international conference on conference on information and knowledge management*. 101–110.

Sherman, G. and M. 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.

Shi, P., H. Bai, and J. Lin. (2020). "Cross-lingual training of neural models for document ranking". In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*. 2768–2773.

Shleifer, S. and A. M. Rush. (2020). "Pre-trained summarization distillation". *arXiv preprint arXiv:2010.13002*.

Singhal, A., C. Buckley, and M. Mitra. (2017). "Pivoted document length normalization". In: *Acm sigir forum*. Vol. 51. No. 2. ACM New York, NY, USA. 176–184.

Soldaini, L. and A. Moschitti. (2020). "The Cascade Transformer: an Application for Efficient Answer Sentence Selection". In: *ACL*.

Song, K., X. Tan, T. Qin, J. Lu, and T.-Y. Liu. (2019). "Mass: Masked sequence to sequence pre-training for language generation". *arXiv preprint arXiv:1905.02450*.

Sordoni, A., Y. Bengio, H. Vahabi, C. Lioma, J. Grue Simonsen, and J.-Y. Nie. (2015). "A hierarchical recurrent encoder-decoder for generative context-aware query suggestion". In: *proceedings of the 24th ACM international on conference on information and knowledge management*. 553–562.

Su, D., Y. Xu, T. Yu, F. B. Siddique, E. J. Barezi, and P. Fung. (2020). "CAiRE-COVID: a question answering and query-focused multi-document summarization system for covid-19 scholarly information management". *arXiv preprint arXiv:2005.03975*.

Su, W., X. Zhu, Y. Cao, B. Li, L. Lu, F. Wei, and J. Dai. (2019). "Vl-bert: Pre-training of generic visual-linguistic representations". *arXiv preprint arXiv:1908.08530*.

Sun, C., A. Myers, C. Vondrick, K. Murphy, and C. Schmid. (2019a). "Videobert: A joint model for video and language representation learning". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 7464–7473.

Sun, S., C. Xiong, Z. Liu, Z. Liu, and J. Bao. (2020). "Joint keyphrase chunking and salience ranking with bert". *arXiv preprint arXiv:2004.13639*.

Sun, Y., S. Wang, Y. Li, S. Feng, X. Chen, H. Zhang, X. Tian, D. Zhu, H. Tian, and H. Wu. (2019b). "Ernie: Enhanced representation through knowledge integration". *arXiv preprint arXiv:1904.09223*.

Tahami, A. V., K. Ghajar, and A. Shakery. (2020). "Distilling Knowledge for Fast Retrieval-based Chat-bots". *arXiv preprint arXiv:2004.11045*.

Tang, H., X. Sun, B. Jin, J. Wang, F. Zhang, and W. Wu. (2021). "Improving Document Representations by Generating Pseudo Query Embeddings for Dense Retrieval". *arXiv preprint arXiv:2105.03599*.

Tang, M., P. Gandhi, M. A. Kabir, C. Zou, J. Blakey, and X. Luo. (2019). "Progress notes classification and keyword extraction using attention-based deep learning models with BERT". *arXiv preprint arXiv:1910.05786*.

Thakur, N., N. Reimers, A. Rücklé, A. Srivastava, and I. Gurevych. (2021). "BEIR: A Heterogeneous Benchmark for Zero-shot Evaluation of Information Retrieval Models". In: *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*. URL: https://openreview.net/forum?id=wCu6T5xFjeJ.

Vaswani, A., N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. (2017). "Attention is all you need". In: *Advances in neural information processing systems*. 5998–6008.

Vinyals, O., A. Toshev, S. Bengio, and D. Erhan. (2015). “Show and tell: A neural image caption generator”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 3156–3164.

Wang, H., X. Wang, W. Xiong, M. Yu, X. Guo, S. Chang, and W. Y. Wang. (2019a). “Self-supervised learning for contextualized extractive summarization”. *arXiv preprint arXiv:1906.04466*.

Wang, W., B. Bi, M. Yan, C. Wu, Z. Bao, J. Xia, L. Peng, and L. Si. (2019b). “Structbert: incorporating language structures into pre-training for deep language understanding”. *arXiv preprint arXiv:1908.04577*.

Wang, X., T. Gao, Z. Zhu, Z. Zhang, Z. Liu, J. Li, and J. Tang. (2021). “KEPLER: A unified model for knowledge embedding and pre-trained language representation”. *Transactions of the Association for Computational Linguistics*. 9: 176–194.

Wu, H., W. Liu, L. Li, D. Nie, T. Chen, F. Zhang, and D. Wang. (2021). “UniKeyphrase: A Unified Extraction and Generation Framework for Keyphrase Prediction”. *arXiv preprint arXiv:2106.04847*.

Wu, S. and M. Dredze. (2019). “Beto, bentz, becas: The surprising cross-lingual effectiveness of BERT”. *arXiv preprint arXiv:1904.09077*.

Wu, Z., J. Mao, Y. Liu, J. Zhan, Y. Zheng, M. Zhang, and S. Ma. (2020). “Leveraging Passage-level Cumulative Gain for Document Ranking”. *Proceedings of The Web Conference 2020*.

Xin, J., R. Nogueira, Y. Yu, and J. J. Lin. (2020a). “Early Exiting BERT for Efficient Document Ranking”. In: *SUSTAINLP*.

Xin, J., R. Tang, J. Lee, Y. Yu, and J. J. Lin. (2020b). “DeeBERT: Dynamic Early Exiting for Accelerating BERT Inference”. In: *ACL*.

Xiong, C., Z. Dai, J. Callan, Z. Liu, and R. Power. (2017a). “End-to-End Neural Ad-hoc Ranking with Kernel Pooling”. *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*.

Xiong, C., Z. Dai, J. Callan, Z. Liu, and R. Power. (2017b). “End-to-end neural ad-hoc ranking with kernel pooling”. In: *Proceedings of the 40th International ACM SIGIR conference on research and development in information retrieval*. 55–64.

Xiong, L., C. Xiong, Y. Li, K.-F. Tang, J. Liu, P. Bennett, J. Ahmed, and A. Overwijk. (2020). "Approximate nearest neighbor negative contrastive learning for dense text retrieval". *arXiv preprint arXiv:2007.00808*.

Xu, S., X. Zhang, Y. Wu, F. Wei, and M. Zhou. (2020). "Unsupervised extractive summarization by pre-training hierarchical transformers". *arXiv preprint arXiv:2010.08242*.

Yamada, I., A. Asai, and H. Hajishirzi. (2021). "Efficient Passage Retrieval with Hashing for Open-domain Question Answering". *arXiv preprint arXiv:2106.00882*.

Yan, M., C. Li, B. Bi, W. Wang, and S. Huang. (2021). "A Unified Pretraining Framework for Passage Ranking and Expansion". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 35. No. 5. 4555–4563.

Yang, W., H. Zhang, and J. J. Lin. (2019a). "Simple Applications of BERT for Ad Hoc Document Retrieval". *ArXiv*. abs/1903.10972.

Yang, Z., Z. Dai, Y. Yang, J. Carbonell, R. Salakhutdinov, and Q. V. Le. (2019b). "XLNet: Generalized Autoregressive Pretraining for Language Understanding". In: *NeurIPS*.

Yates, A., R. Nogueira, and J. Lin. (2021). "Pretrained Transformers for Text Ranking: BERT and Beyond". In: *SIGIR '21: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, Canada, July 11-15, 2021*. Ed. by F. Diaz, C. Shah, T. Suel, P. Castells, R. Jones, and T. Sakai. ACM. 2666–2668. DOI: 10.1145/3404835.3462812. URL: https://doi.org/10.1145/3404835.3462812.

Yin, D., Y. Hu, J. Tang, T. Daly, M. Zhou, H. Ouyang, J. Chen, C. Kang, H. Deng, C. Nobata, J.-M. Langlois, and Y. Chang. (2016). "Ranking Relevance in Yahoo Search". *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

Yin, W. and Y. Pei. (2015). "Optimizing sentence modeling and selection for document summarization". In: *Twenty-Fourth International Joint Conference on Artificial Intelligence*.

Yu, S., Z. Liu, C. Xiong, T. Feng, and Z. Liu. (2021). "Few-Shot Conversational Dense Retrieval". *arXiv preprint arXiv:2105.04166*.

Zhai, C. (2008). "Statistical Language Models for Information Retrieval: A Critical Review". *Found. Trends Inf. Retr.* 2: 137–213.

Zhai, C. and J. Lafferty. (2004). "A study of smoothing methods for language models applied to information retrieval". *ACM Transactions on Information Systems (TOIS)*. 22(2): 179–214.

Zhan, J., J. Mao, Y. Liu, J. Guo, M. Zhang, and S. Ma. (2021a). "Jointly Optimizing Query Encoder and Product Quantization to Improve Retrieval Performance". *arXiv preprint arXiv:2108.00644*.

Zhan, J., J. Mao, Y. Liu, J. Guo, M. Zhang, and S. Ma. (2021b). "Learning Discrete Representations via Constrained Clustering for Effective and Efficient Dense Retrieval". *arXiv preprint arXiv:2110.05789*.

Zhan, J., J. Mao, Y. Liu, J. Guo, M. Zhang, and S. Ma. (2021c). "Optimizing Dense Retrieval Model Training with Hard Negatives". *arXiv preprint arXiv:2104.08051*.

Zhan, J., J. Mao, Y. Liu, M. Zhang, and S. Ma. (2020a). "Learning To Retrieve: How to Train a Dense Retrieval Model Effectively and Efficiently". *arXiv preprint arXiv:2010.10469*.

Zhan, J., J. Mao, Y. Liu, M. Zhang, and S. Ma. (2020b). "RepBERT: Contextualized Text Embeddings for First-Stage Retrieval". *CoRR*. abs/2006.15498. arXiv: 2006.15498. URL: https://arxiv.org/abs/2006.15498.

Zhang, H., H. Shen, Y. Qiu, Y. Jiang, S. Wang, S. Xu, Y. Xiao, B. Long, and W.-Y. Yang. (2021). "Joint Learning of Deep Retrieval Model and Product Quantization based Embedding Index". *arXiv preprint arXiv:2105.03933*.

Zhang, H., J. Xu, and J. Wang. (2019a). "Pretraining-based natural language generation for text summarization". *arXiv preprint arXiv:1902.09243*.

Zhang, J., Y. Zhao, M. Saleh, and P. Liu. (2020). "Pegasus: Pre-training with extracted gap-sentences for abstractive summarization". In: *International Conference on Machine Learning*. PMLR. 11328–11339.

Zhang, X., F. Wei, and M. Zhou. (2019b). "HIBERT: Document level pre-training of hierarchical bidirectional transformers for document summarization". *arXiv preprint arXiv:1905.06566*.

Zhang, Z., X. Han, Z. Liu, X. Jiang, M. Sun, and Q. Liu. (2019c). "ERNIE: Enhanced language representation with informative entities". *arXiv preprint arXiv:1905.07129*.

Zheng, Z., K. Hui, B. He, X. Han, L. Sun, and A. Yates. (2020). "BERT-QE: contextualized query expansion for document re-ranking". *arXiv preprint arXiv:2009.07258*.

Zheng, Z., K. Hui, B. He, X. Han, L. Sun, and A. Yates. (2021). "Contextualized query expansion via unsupervised chunk selection for text retrieval". *Information Processing & Management*. 58(5): 102672.

Zhong, M., P. Liu, Y. Chen, D. Wang, X. Qiu, and X. Huang. (2020). "Extractive summarization as text matching". *arXiv preprint arXiv:2004.08795*.

Zhong, M., P. Liu, D. Wang, X. Qiu, and X. Huang. (2019). "Searching for effective neural extractive summarization: What works and what's next". *arXiv preprint arXiv:1907.03491*.

Zhou, Y., Z. Dou, B. Wei, R. Xie, and J.-R. Wen. (2021a). "Group Based Personalized Search by Integrating Search Behaviour and Friend Network". In: *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval. SIGIR '21*. New York, NY, USA. 92–101.

Zhou, Y., Z. Dou, and J.-R. Wen. (2020). "Encoding History with Context-aware Representation Learning for Personalized Search". In: *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1111–1120.

Zhou, Y., Z. Dou, Y. Zhu, and J.-R. Wen. (2021b). "PSSL: Self-supervised Learning for Personalized Search with Contrastive Sampling". In: *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. 2749–2758.

Zhu, H., L. Dong, F. Wei, B. Qin, and T. Liu. (2019). "Transforming wikipedia into augmented data for query-focused summarization". *arXiv preprint arXiv:1911.03324*.

Zhu, Y., J.-Y. Nie, Z. Dou, Z. Ma, X. Zhang, P. Du, X. Zuo, and H. Jiang. (2021). "Contrastive Learning of User Behavior Sequence for Context-Aware Document Ranking". In: *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. 2780–2791.

Zou, L., S. Zhang, H. Cai, D. Ma, S. Cheng, D. Shi, S. Wang, Z. Cheng, and D. Yin. (2021). "Pre-trained Language Model based Ranking in Baidu Search". *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*.

Zou, Y., X. Zhang, W. Lu, F. Wei, and M. Zhou. (2020). "Pre-training for Abstractive Document Summarization by Reinstating Source Text". *arXiv preprint arXiv:2004.01853*.