Pre-training Methods in Information Retrieval

Suggested Citation: Yixing Fan[*], Xiaohui Xie[*], Yinqiong Cai, Jia Chen, Xinyu Ma, Xiangsheng Li, Ruqing Zhang and Jiafeng Guo[⋆] (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

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] and Jiafeng Guo⋆[1]

[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*

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 the user's information need. In recent years, 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

* Yixing Fan and Xiaohui Xie contributed equally.

⋆ Corresponding authors.

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

task of IR. Recently, a large number of works, which are dedicated to the application of PTMs in IR, have been introduced to promote the retrieval performance. Considering the rapid progress of this direction, this survey aims to provide a systematic review of pre-training methods in IR. To be specific, we present an overview of PTMs applied in different components of an 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 highlight several promising directions, with the hope of inspiring and facilitating more works on these topics for future research.

1

Introduction

Information retrieval (IR) is a fundamental task in many real-world applications, such as Web search, question answering systems, digital libraries, 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 more than one relevant resource, the returned result is often organized as a ranked list of documents (e.g., Web pages, answers, or responses) 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 introduced and studied, including vector space models (Salton *et al.*, 1975), probabilistic ranking models (Robertson and Jones, 1976), and learning to rank (LTR) models (Li, 2014). These methods have been successfully applied in many different IR applications, such as Web search engines like Google, news recommender systems like Toutiao, community question answering platforms 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 (Craswell *et al.*, 2017) (i.e., neural information retrieval). Different from previous non-neural ranking models that rely on elaborately-designed features and manually-designed functions, neural ranking models can automatically learn low-level dense representations from data as ranking features. 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 (Dehghani *et al.*, 2017b).

In recent years, 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 with self-supervised language modeling, e.g., predicting the probability of a masked token, and then adapt the pre-trained model to downstream tasks by introducing a small number of additional parameters and fine-tuning them with some task-specific objectives. As is demonstrated in recent works (Peters *et al.*, 2018; Howard and Ruder, 2018), these pre-trained models are able to capture a decent amount of linguistic knowledge as well as factual knowledge, which are beneficial for downstream tasks and can avoid learning such knowledge from scratch. Moreover, with the increasing amount of computational power and the emergence of the Transformer architecture (Vaswani *et al.*, 2017), we can further improve the capacity of pre-trained models by updating the parameter scale, e.g., from million-level to billion-level (e.g., BERT (Devlin *et al.*, 2019) and GPT-3 (Brown *et al.*, 2020)) and even trillion-level (e.g., Switch-Transformers (Fedus *et al.*, 2021)). Both of these are desirable properties for modeling the relevance in IR. On one hand, pre-trained embeddings, which are learned on huge textual corpus with self-supervised modeling objectives, are able to capture intrinsic semantics inside queries and 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. Owing to these potential benefits, we have witnessed explosive growth of research interest in exploiting PTMs in IR (Onal *et al.*, 2017; Lin *et al.*, 2021a). Note that in this survey, we focus on PTMs in text retrieval, which is central to IR. Readers who are interested in PTMs in content-based image retrieval or multi-modal retrieval could refer to (Dubey, 2020; Fei *et al.*, 2021).

Up to now, numerous studies have been devoted to the application of PTMs in IR. In academia, researchers have carried out a variety of innovation and initiative in the usage of PTMs in IR. For example, earlier attempts tried to leverage pre-trained word embeddings to promote ranking models, and have achieved some notable results (Onal *et al.*, 2017). More recent works proposed to improve existing pre-trained models by either reforming the model architecture (MacAvaney *et al.*, 2020; Khattab and Zaharia, 2020; Gao and Callan, 2021a) or considering novel pre-training objectives (Chang *et al.*, 2020; Ma *et al.*, 2021b; Ma *et al.*, 2021c), which better meet the requirements of IR. Meanwhile, in industry, Google's October 2019 blog post[1] and Bing's November 2019 blog post[2] 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[3] 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 the current literature, highlight advantages and limitations of 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, ICLR, WSDM, CIKM, AAAI, ACL, 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 2016 to 2021. There exists some previous works discussing related topics. For example, both Onal *et al.* (2017) and Guo *et al.* (2020) reviewed the landscape of neural retrieval models used in three major IR tasks. They also discussed early usage of pre-trained embeddings in neural ranking models, but did not cover every aspect of PTMs in IR. Guo *et al.* (2022) reviewed semantic models for the first-stage retrieval, including early semantic retrieval models,

[1] https://www.blog.google/products/search/search-language-understanding-bert/

[2] https://azure.microsoft.com/en-us/blog/bing-delivers-its-largest-improvement-\in-search-experience-using-azure-gpus/

[3] https://microsoft.github.io/msmarco/#docranking

neural retrieval models, and retrieval models based on PTMs. More recently, Lin *et al.* (2021a) provided a thorough survey of transformer-based models for IR, which reviews existing literature on the application of pre-trained contextual embedding in text ranking. Different from these works, we make a comprehensive overview of PTMs applied in IR, including the usage of pre-trained word embeddings as well as the application of pre-trained transformers. More specifically, we reviewed the application of PTMs in different components of an 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 research 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 PTMs can be beneficial for IR.

2.1 A Hierarchical View of 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. rel refers to the relevance estimation model which calculate the relevance scores s_{ij} for each (q_i, d_j) pair. R_q denotes returned search results against an issued query q.

2.1.1 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

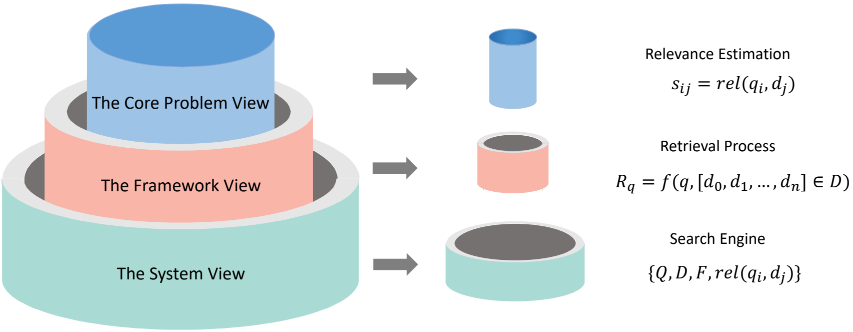


Figure 2.1: A Hierarchical View of IR

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 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 the better the 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 sum of contributions from each query term that appears in the document is used to derive the relevance score. Among these models, BM25 (Robertson *et al.*, 1994) is shown to be effective and is still 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 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 techniques 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 consider individual documents and regard the retrieval problem as classification or regression problem. Example models include PRank (Perceptron Ranking) (Crammer and Singer, 2001) and McRank (Li *et al.*, 2007). 2) Pairwise approaches which take pairs of documents into consideration. For example, RankNet (Burges *et al.*, 2005) is a pairwise method which adopts Cross Entropy as loss function in learning and RankSVM (Herbrich, 1999) which performs ranking as a pairwise classification problem 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 generality, these models can be grouped into representation-focused models, interaction-focused models, and mixed models. 1) Representation-focused models aims at learning dense vector representations of queries and documents independently. Then metrics such as cosine similarity and inner products are used to calculate the "distance" between queries and documents to estimate 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 embedding 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.*, 2016) 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.*, 2019) for example. For more detailed information please refer to these earlier surveys (Onal *et al.*, 2017; Guo *et al.*, 2020) on NeuIR models for IR

2.1.2 The Framework View of IR

Given a document collection D, the aim of IR is to provide a search result list R_q 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 (Frieder *et al.*, 2000). 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 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.*, 1994) 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 later-stage re-rankers. Compared to later-stage re-rankers, early-stage re-rankers will focus more on efficiency but will pay more attention to effectiveness than retrievers. Since the number of documents considered by later-stage re-rankers is small, later-stage re-rankers will focus more on effectiveness. Conventional re-ranking models include learning to rank models (e.g., RankNet (Burges *et al.*, 2005) and LambdaMart (Burges *et al.*, 2006)) and neural models (e.g., DRMM (Guo *et al.*, 2016) and Duet (Mitra *et al.*, 2017)).

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

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

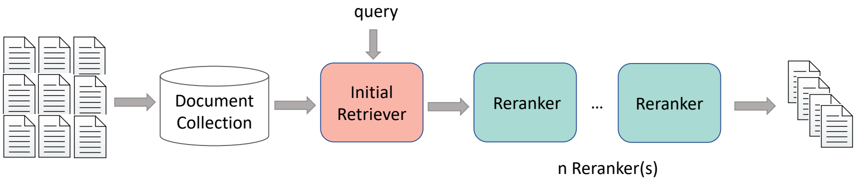


Figure 2.2: The retrieval architecture. According to the number of re-rankers, this retrieval process can be defined as Single-stage Retrieval ($n = 0$), Two-stage Retrieval ($n = 1$) and Multi-stage Retrieval ($n \geq 2$).

2.1.3 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-spelt. 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 given search query. Without the document index, the search system would need to scan every document in the corpus, which is time-consuming and requires considerable 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 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 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 that appear 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 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 (Manning *et al.*, 2008).

- **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 (Mitra and Craswell, 2018).

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 to rank models, while neural search systems adopt more dense retrieval models and neural ranking models.

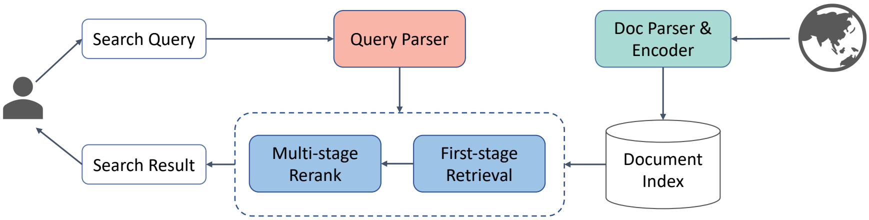


Figure 2.3: The framework of a practical search system.

2.2 A Brief Overview of PTMs in IR

Deep learning models are data-hungry. Especially for models with a massive number of parameters, large datasets are needed to fully learn model parameters and circumvent overfitting issues. However, building a large-scale labeled dataset for IR is a laborious, expensive and time-taking task. In contrast, constructing large-scale yet unlabeled corpora (e.g., crawled web pages and search logs) is much easier. Thus, an intuitive way is to employ PTMs to exploit the corpora to learn a better initialization of model parameters. Then, the workflow becomes: 1) PTMs are first applied to learn either good representations of texts or better interaction between text-pairs based on unlabeled datasets; 2) the learned representations/interactions are then fine-tuning and used for downstream tasks. Specifically, depending on the target downstream task, there exist different options for the fine-tuning: 1) Full fine-tuning: fine-tuning all weights with the data from the downstream task; 2) Partial fine-tuning: fine-tuning partial weights that are specific to the downstream task while freezing the other weights; 3) Freezing the weights: using the representation from the frozen weight to solve the downstream task. Existing works show that learned representations or interactions extracted from the PTMs are beneficial for many IR tasks such as document retrieval and re-ranking (Guo *et al.*, 2016; Lin *et al.*, 2021a). In this Section, we briefly overview typical PTMs in IR and introduce how they benefit IR in different stages of the search system. The purpose of this section is to help readers to gain basic knowledge of pre-training methods designed for IR tasks.

The development of PTMs in IR has roughly gone through two phases. During the 2010s, in the first phase, word embedding methods have been investigated to develop meaningful representations of words. While recently, in the second phase, transformer-based methods are proposed to gain better representations or interactions of texts by considering more sophisticated model structures and pre-training objectives. We briefly overview these two methods and their relationship to IR.

2.2.1 Word Embedding Methods

An embedding refers to a representation of items in a new space where the properties of items and the relationship between these items are preserved. Then the relatedness of items can be computed based on the notion of similarity in this new space. In that regard, if the item representations are close to one another means that those items are close to one another. Word embedding methods learn word representation by setting up an unsupervised prediction task which enables pre-training in a large corpus before using the representation in downstream tasks. Specifically, the objective is to have words with similar contexts occupy close spatial positions in the new space. This section briefly overviews classical word embedding methods and their usages in IR tasks. Classical word embedding methods can be categorized into the following groups:

- **Word2vec**: In Word2vec approaches (Mikolov *et al.*, 2013a; Mikolov *et al.*, 2013b; Mikolov *et al.*, 2013c), the word embedding of a term is learned by considering its neighbours within a fixed size window over the text. There are two architectures, i.e., skip-gram and continuous bag-of-words (CBOW). Both architectures apply a shallow neural model with one hidden states. For the skip-gram architecture, given a center word, the model learns to predict the most likely words in a fixed-sized window around it. For the CBOW architecture, in contrast, the model learns to predict the center word based on the context words. Since the skip-gram architecture creates more training samples from the same window of text, it trains slower than the CBOW model during training phase (Mikolov *et al.*, 2013a).
- **GloVe**: Pennington *et al.* (2014) proposed GloVe that generates

global vectors for word representation. Unlike training on individual term-neighbor pairs as in word2vec approaches, GloVe performs training on aggregated global word-word co-occurrence statistics from a corpus. Different from applying a feedforward neural model, GloVe constructs a word-context matrix, i.e., for each “word”, how frequently we see this word in some “context” can be counted. Then the matrix factorization technique is utilized to yield a lower-dimensional matrix (embedding matrix) where each row refers to a vector representation (word embedding) for a corresponding word.

- **Paragraph2vec**: Paragraph2vec (Le and Mikolov, 2014), also known as Doc2vec, is another widely used technique that creates an embedding of a generic block of text, such as sentences, paragraphs and documents. Expanding upon the Word2vec, Paragraph2vec adds another vector that represents the paragraph ID to the input. In that regard, while training the word embedding, the numeric representation of the paragraph can also be obtained. In the context of IR tasks, Ai *et al.* (2016a) and Ai *et al.* (2016b) proposed a number of changes tailored for IR to the original Paragraph2vec, i.e., document frequency based negative sampling and document length based regularization.

Unsupervised and pre-trained word embeddings can be incorporated into IR models and enhance the performance of these models due to their great abilities in capturing semantic and syntactic properties of the input texts. **word embeddings are used to refine term weighting schemes in the inverted index**. For example, Zheng and Callan (2015) proposed DeepTR that leverages pre-trained word embeddings learned by the CBOW-based Word2vec. DeepTR can estimate the term importance and replaces classical term weighting schemes, such as Term Frequency (TF), in the inverted index so as to improve the retrieval performance. Moreover, **word embeddings are applied to better estimate the matching levels of queries and documents**. For example, Zamani *et al.* (2018b) proposed SNRM that learns sparse representation for each query and document based on pre-trained word embeddings to better capture semantic relationships between them.

They then constructed an inverted index based on the learned sparse representation which enhances the performance of retrieval. Gysel *et al.* (2018) proposed the Neural Vector Space Model (NVSM) that is a pre-trained word embeddings method tailored for IR. In the NVSM paradigm, they learn low-dimensional representations of words and documents from scratch using gradient descent and rank documents according to their similarity with query representations that are composed of word representations. Furthermore, **word embeddings are adopted to benefit crucial IR-related tasks, e.g., query suggestion and document summarization**. For example, Dehghani *et al.* (2017a) used word2vec as an input to encode queries and then feed the query representations into a customized sequence-to-sequence model to deal with the session-based query suggestion problem. Yin and Pei (2015) bulit a CNN-based summarizer, named DivSelect+CNNLM, to enhance the performance of the extractive summarization. Specifically, the CNNLM module is pre-trained on a large corpus to learn better sentence representations by capturing more internal semantic features.

2.2.2 Transformer-based Methods

Although word embedding methods are demonstrated to be beneficial for IR tasks, they can not deal with the context-dependent nature of words and the issue of polysemous. This motivates attempt at constructing pre-training methods that can learn context-aware representations of words or interactions between words. Among them, Transformer (Vaswani *et al.*, 2017) is a successful instance and has been widely adopted in IR scenarios. This section briefly overviews typical transformer-based methods, including the structures and pre-training objectives. We also provide examples of using transformer-based methods in IR tasks.

Vaswani *et al.* (2017) proposed transformer, an encoder-decoder architecture that consists of stacked self-attention and point-wise, fully connected layers and supplement modules including positional embeddings, layer normalization and residual connections. Specifically, in the encoding phase, the transformer first calculates an attention score by comparing a given word with each other word in the input sequence. The attention score indicates that how much each of the other words should contribute to the next representation of the given word. Transformer then utilizes these attention scores to compute a weighted average of the representations of all the words in the input sequence. The attention mechanism of the decoding phase is similar to the encoding phase. The difference is that the attention mechanism in the decoding phase only decodes one representation from left to right at a time and each step of the decoding phase takes into account results decoded in the previous step. Due to the parallel modeling capabilities of the self-attention mechanism, transformer is able to train big models with extensive parameters using advanced computing devices. In that regard, transformer has served as the backbone neural structure for the subsequently derived PTMs.

GPT (Radford *et al.*, 2018) and BERT (Devlin *et al.*, 2019) are two landmark models of transformer-based pre-training methods. Among them, GPT uses auto-regressive language modeling as the pre-training objective. In particular, the objective is to maximize the conditional probabilities of all the words in the context of their corresponding previous words. Hence, GPT is good at generation tasks. And BERT applies auto-encoding language modeling as the pre-training objective and focus more on language understanding and discriminative tasks. More specifically, two pre-training objectives word together to optimize the parameters of BERT in the pre-training phase: 1) Masked language modeling (MLM): tokens are randomly masked with a special token [MASK] and the objective is to predict words at the masked positions in the context of other words; 2) Next sentence prediction (NSP): the objective is to predict whether two sentences are coherent with a binary classifier.

Due to their great ability on capturing polysemous disambiguation, syntactic and lexical structures, also the factual knowledge contained in the text, GPT, BERT and their successors have achieved success in IR scenarios. **Transformer-based methods are used 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.*, 2021)) and interaction-focused (e.g., MonoBERT (Nogueira and Cho, 2019),

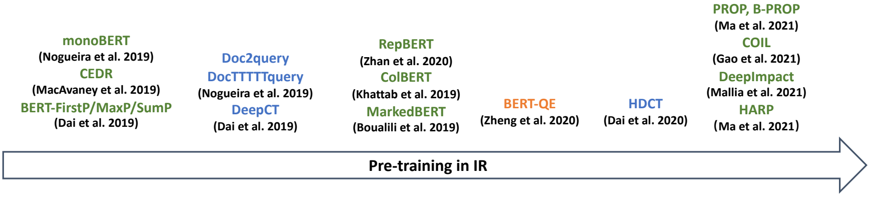


Figure 2.4: Recent PTMs in IR. "Orange", "Green" and "Blue" refer to the "Query Parser", "Retrieval and Rerank", and "Doc Parser & Encoder" stages for which PTMs target respectively.

CEDR (MacAvaney *et al.*, 2019) 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 the two encoders are then fed into a "similarity" function to obtain the relevance score. MonoBERT (interaction-focused) takes the concatenation of the 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. Moreover, **transformer-based methods also considers the trade-off between efficiency and effectiveness according to the stages (retrieval or reranking) they targets.** 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. In contrast for the re-ranking stage, PTMs need to deal with a small set of documents and capture more fine-grained relevance signals. For example, CEDR (MacAvaney *et al.*, 2019) 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. **Different transformer-based methods 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 MeshBART (Chen and Lee, 2020) leverages user behavioral patterns 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 the predicted term weights are used 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, the "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 the Retrieval Component

Traditional search engines rely on term-based retrieval models like BM25 (Robertson and Zaragoza, 2009) for effective and efficient retrieval. Recently, with the rapid progress in representation learning (Bengio *et al.*, 2013) and pre-training methods (Devlin *et al.*, 2019; Yang *et al.*, 2019; Radford *et al.*, 2019), PTMs-based retrieval models have become the popular paradigm to improve retrieval effectiveness. While equipped with PTMs, retrieval models have achieved great progress in terms of effectiveness (Yan *et al.*, 2021; Karpukhin *et al.*, 2020). In this section, we briefly review pre-training methods applied in the retrieval component. Firstly, we give a comprehensive summary of pre-trained retrieval models in terms of model structures. Then, we discuss several challenges and promising topics in terms of the learning of retrieval models.

3.1 Basic Model Structure

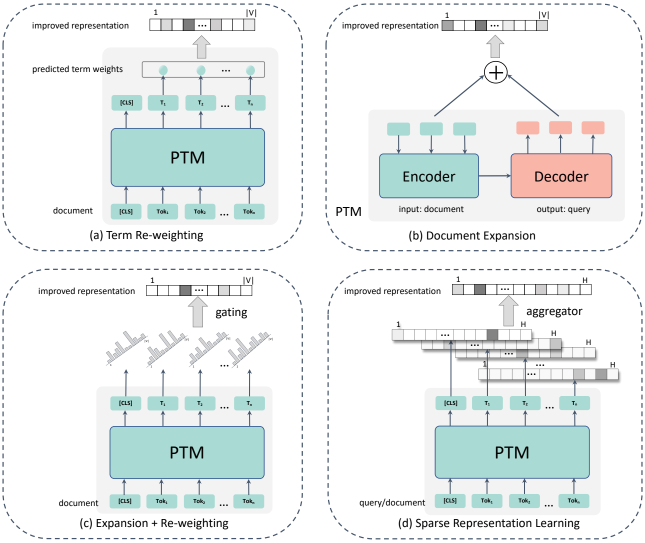
From the perspective of representation type and index mode, PTMs-based retrieval models can be devided into three categories (Guo *et al.*, 2022): 1) Sparse Retrieval Models: improve retrieval by obtaining semantic augmented sparse representations and index them with the inverted index for efficient retrieval; 2) Dense Retrieval Models: project input texts (i.e., queries and documents) into standalone dense representations and turn to approximate nearest neighbor search algorithms for fast retrieval; 3) Hybrid Retrieval Models: build sparse and dense retrieval models concurrently to absorb merits of both for better retrieval performance.

3.1.1 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 framework, queries and documents are represented with high-dimensional sparse embeddings so that the inverted index can be still used for efficient retrieval (Dai and Callan, 2019a; Bai *et al.*, 2020).

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.

Term Re-weighting One of the most direct ways to improve the term-based retrieval is to measure term weights with contextual semantics, instead of term frequency (TF) (Figure 3.1 (a)). Originally, there have been works utilizing pre-trained word embeddings to estimate term importance. Earliest, Zheng and Callan (2015) leveraged term weights estimated by pre-trained word embeddings to replace TF in the inverted index to improve the retrieval effectiveness. Later, Frej *et al.* (2020) utilized FastText (Bojanowski *et al.*, 2017) to estimate the IDF field in the inverted index. For the above models, the pre-trained word embeddings could be fixed or fine-tuned during the retrieval models training. Recently, with the development of pre-trained models, there are also explorations to utilize them to estimate term weights. For example, Dai and Callan (2020a) used BERT to obtain contextualized token embeddings, and then mapped them to term weights, instead of TF, to build the inverted index. Later, Dai and Callan (2020b) adapted



(a) Term Re-weighting

(b) Document Expansion

(c) Expansion + Re-weighting

(d) Sparse Representation Learning

Figure 3.1: Four architectures of sparse retrieval models.

DeepCT (Dai and Callan, 2020a) to estimate term weights for long documents and proposed the HDCT model. It firstly estimates passage-level term weights as the DeepCT does, and then uses a weighted sum to combine them into document-level term weights.

Document Expansion Besides explicitly predicting term weights, augmenting the document with semantically related terms is another practical method (Figure 3.1 (b)). Based on this, the vocabulary mismatch problem can be alleviated to some extent, and elite terms in the document are promoted at the same time. In fact, compared with extensive works on query expansion based on PTMs, document expansion are less popular in the IR field. Different from early methods that expand documents 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) firstly fine-tuned a pre-trained language model T5 (Raffel *et al.*, 2020) with relevant query-document pairs. The learned model generates multiple queries for each document and appends them to the original document. Then, they used BM25 to retrieve relevant documents based on the expanded document collection. Later, based on the assumption that document ranking and document expansion tasks share certain inherent relations and can benefit from each other, Yan *et al.* (2021) used the document ranking task to enhance the training of document expansion task. They firstly pre-trained the Transformer encoder-decoder architecture (Vaswani *et al.*, 2017), where the encoder is pre-trained to support document re-ranking and the decoder is pre-trained for query generation. Then, they conducted a joint fine-tuning process, where a mini-batch is constructed 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.

Expansion + Re-weighting Based on the above two methods, a more optimal method is to combine the idea of term re-weighting and document expansion, learning term weights in the whole vocabulary instead of existing tokens in the document (Figure 3.1 (c)). For example, SparTerm (Bai *et al.*, 2020) predicts the term importance distribution in the vocabulary space based on contextual token embeddings got by BERT. Based on this, it re-weights existing and expand terms simultaneously. Moreover, it includes a gating controller to ensure the sparsity of the final representation. Later, Formal *et al.* (2021) proposed SPALDE to improve SparTerm (Bai *et al.*, 2020), which used a saturate function to prevent some terms from dominating the representation and employs a *FLOPS* loss to enable the 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 expand documents firstly, and then uses BERT to estimate term importance for appeared terms.

Sparse Representation Learning Different from the above methods to improve document representations in explicit symbolic space, sparse representation learning methods learn sparse embeddings for queries and documents in the latent word space (Figure 3.1 (d)). SNRM (Zamani *et al.*, 2018b) is the pioneer to learn sparse representations for ad-hoc retrieval. Based on the pre-trained word embeddings, SNRM learns standalone sparse representations for each query and document to capture semantic relationships between them, which shows better retrieval effectiveness over baselines. Recently, Jang *et al.* (2021) proposed UHD-BERT, which learns extremely high dimensional representations with controllable sparsity based on pre-trained language models. More specifically, it firstly obtains dense token embeddings for queries/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 embeddings and get the sparse token embeddings. Finally, it generates the sparse query/document representation by token-wise max pooling. Besides, Yamada *et al.* (2021) integrated the learning-to-hash technique into DPR (Karpukhin *et al.*, 2020) to represent input texts with binary codes. BPR is learned with a multi-task objective, which trains the BERT-based dual-encoder and the hash function in an end-to-end manner. Based on the binary codes of queries and documents, BPR drastically reduces the memory cost of the document index and obtains comparable accuracy on two benchmarks.

3.1.2 Dense Retrieval Models

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

Dense retrieval models usually consist of two encoders to learn standalone dense embeddings for queries and documents independently. Then, a simple matching function (e.g., dot product or cosine similarity) is used to calculate the relevance scores based on the learned representations. In this way, the basic architecture of dense retrieval models can be formulated as:

$$rel(q, d) = f(\phi_{PTM}(q), \varphi_{PTM}(d)), \tag{3.1}$$

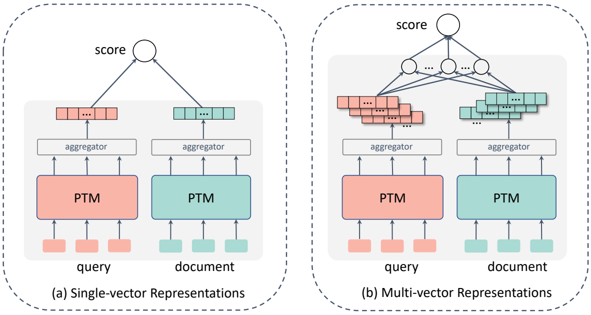

score
aggregator
PTM
query
document
(a) Single-vector Representations
score
aggregator
PTM
query
document
(b) Multi-vector Representations

Figure 3.2: Basic architectures of dense retrieval models.

where ϕ_{PTM} and φ_{PTM} are query and document encoders based on pre-training methods, and f is the similarity function. In the literature, 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.

Single-vector Representation Initially, some works used simple heuristic functions to aggregate pre-trained word embeddings and obtained dense representations for queries and documents. For example, Clinchant and Perronnin (2013) presented a document representation model based on pre-trained word embeddings. They used the fisher kernel framework to transform word embeddings into a high-dimensional space and then aggregated them to generate the document representation. Afterwards, Gillick *et al.* (2018) obtained query and document representations with the average of pre-trained word embeddings. The surprising experimental results indicate that dense retrieval is a practical alternative to the symbolic-based retrieval models. Besieds, Gysel *et al.* (2018) and Agosti *et al.* (2020) proposed word-embedding learning methods tailored for IR (see Section 6 for details). However, it is easy to find that obtaining query/document representations by directly aggregating word embeddings would lose contextual semantics and word orders information. To address this problem, Le and Mikolov (2014) proposed the Paragraph Vector (PV) algorithm to learn fixed-length representations from variable-length texts. Later, Ai *et al.* (2016b) found the unstable performance and limited improvements of PV representations for ad-hoc retrieval and produced modifications to it for IR tasks.

Except for obtaining dense query/document representations based on pre-trained embeddings, 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. One of the representatives that apply pre-trained models for dense retrieval is DPR (Karpukhin *et al.*, 2020), which is proposed for OpenQA tasks. DPR learns dense embeddings for queries and passages with two independent BERT-based encoder. Then, relevance scores are calculated with the inner product operation between query and document representations. The results on several OpenQA datasets show that DPR outperforms BM25 and is beneficial for the downstream QA performance. For ad-hoc retrieval tasks, Zhan *et al.* (2020b) proposed RepBERT to replace BM25 for the retrieval component. 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. Similarly, the PTMs-based dense retrieval method also improves conversational search. For example, Yu *et al.* (2021) presented ConvDR to learn contextualized BERT embeddings for multi-turn conversational queries and documents respectively, and then retrieves relevant documents using dot products. Another approach to building a strong dense retriever is to distill the learned knowledge from a more complex model (Tahami *et al.*, 2020; Lin *et al.*, 2021b; Choi *et al.*, 2021; Hofstätter *et al.*, 2020). For example, Tahami *et al.* (2020) utilized the knowledge distillation (KD) technique to distillate the BERT-based cross-encoder network to the dual-encoder model, which heavily increases the retrieval effectiveness.

Multi-vector Representation Besides learning a single global representation for queries and documents, another approach is to obtain multiple vectors for them. A natural method is to take pre-trained word embeddings as term-level representations for queries and documents. Earliest, Kenter and Rijke (2015) proposed to rely only on pre-trained word embeddings for short texts retrieval. They took the cosine similarity between the query word embedding document word embedding to replace the TF field in BM25 for retrieval, which shows better performance than baselines. Later, Mitra *et al.* (2016) proposed to retain dual word embedding spaces. Based on the learned pre-trained word2vec embedding model, query words are mapped into the input space and document words are mapped into the output space. The final relevance score is calculated with aggregated cosine similarities between all query-document word pairs.

Except for the pre-trained word embeddings, there are also a number of works that employ pre-trained models to learn query/document representations for IR. ColBERT (Khattab and Zaharia, 2020) generates contextualized term embeddings for queries and documents with a BERT-based dual-encoder, and then employs the MaxSim operator to obtain the matching score. Later, Gao *et al.* (2021a) proposed a similar method, but only calculating similarities between exactly matched terms for queries and documents in the MaxSim operator. Besides, an alternative way is to employ different encoders for queries and documents based on the heterogeneity between documents and queries. For example, Luan *et al.* (2021) proposed ME-BERT, which takes the contextualized embedding of CLS as the single-vector query representation and the first m contextualized token embeddings as the multi-vector document representation. Finally, the largest inner product between each document vector with the query vector is token as the relevance score. Recently, Tang *et al.* (2021) proposed a novel multi-vector representation method, which clusters BERT-based document term embeddings with k-means to generate multiple representations for each document. Experimental results show that the model can improve retrieval results significantly on several QA datasets .

3.1.3 Hybrid Retrieval Models

Sparse retrieval models take a (latent) word as the unit of representations, which can calculate the matching score based on exact matching signals. On the other hand, dense retrieval methods learn dense embeddings for queries and documents and the relevance is evaluated with soft matching signals. To benefit from both of them, hybrid retrieval models learn sparse and dense representations for queries and docu-

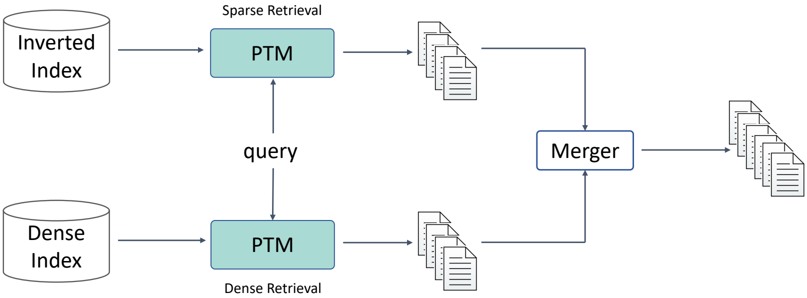


Figure 3.3: The architecture of hybrid retrieval models.

ments simultaneously and calculate the final relevance scores with a merging method (Figure 3.3).

To begin with, there are a number of works proposing to combine pre-trained word embeddings with term-based models for the retrieval component. For example, Vulić and Moens (2015) combined word embeddings with the language model for monolingual and bilingual retrieval and obtained better results. Besides, Roy *et al.* (2016a) also proposed to inject pre-trained word embeddings into the standard query likelihood model (QL) for document retrieval. However, most of these works got the conclusion that only relying on pre-trained word embeddings to build the retrieval model always shows poor performance, unless combining it with the term-based retrieval method.

With the boosting development of pre-trained models, they are naturally combined with term-based models to enhance retrieval effectiveness. Seo *et al.* (2019) proposed to learn dense and sparse representations for each phrase in the collection concurrently for OpenQA tasks, where the dense vector is constructed by BERT-based embeddings, and the sparse embedding is the tf-idf representation of the phrase. Afterwards, Lee *et al.* (2020) proposed to replace the TF-based sparse representation in DenSPI (Seo *et al.*, 2019) with a learned contextual sparse representation based on BERT. 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 (Lin and Ma, 2021; Luan *et al.*, 2021). For example, Luan *et al.* (2021) proposed to linearly combine BM25 and ME-BERT (Luan *et al.*, 2021) to produce strong performance. There are also works using more sophisticated merging methods. For example, Kuzi *et al.* (2020) leveraged a hybrid approach (BM25 + DE-BERT) with RM3 (Jaleel *et al.*, 2004) as the merger for document retrieval, and Gao *et al.* (2020c) proposed CLEAR to learn the BERT-based dense retriever with the residual of a sparse retrieval model (BM25).

3.2 Advanced Topics

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

3.2.1 Negative Sampling Strategy

The negative sampling strategy is a key factor for determining the performance of learned retrieval models. Generally, hard negative examples are considered as informative negatives, because they can improve the ability of the model to differentiate similar examples. Thus, how to integrate hard negatives into the learning of PTMs-based retrievers is a widely concerned topic.

One of exemplary methods is the ANCE training method (Xiong *et al.*, 2021), which firstly warms up the RoBERTa-based (Liu *et al.*, 2019) dense retriever with BM25 negatives, and then continues the dual-encoder training with the periodically refreshed ANN index for hard negative sampling. Experimental results indicate that ANCE elevates dense retrievers and convincingly surpasses baselines on several benchmarks. Later, Zhan *et al.* (2020a) and Zhan *et al.* (2021b) 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.*, 2021) or STAR (Zhan *et al.*, 2021b)). Then, at each training step, they performed full retrieval based on the fixed document index and updated the query encoder with top retrieved documents as negatives. Experimental results 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) argued that previous methods select the training batch with random queries, making in-batch negatives with little information for dense retrievers training. Based on this observation, they proposed to train dense retrievers with TAS-Balanced batches, which composes training batches with topic-aware query sampling and margin-balanced negative sampling.

3.2.2 Joint Learning with Other Components

To improve retrieval performance, PTMs-based retrieval models can be learned jointly with the index module. Besides, for different applications, the retrieval component can be learned with downstream components end-to-end, e.g., re-rankers for ad-hoc retrieval and readers for OpenQA.

Joint Learning with Index As mentioned above, efficiency is one of the core considerations for the retrieval component. To support rapid online search, retrieval systems usually build an index for all documents in the collection. Specially, for dense retrieval methods described in Section 3.1.2, they usually rely on ANN search algorithms (Aumüller *et al.*, 2020; Echihabi *et al.*, 2019; Li *et al.*, 2020c) to perform efficient retrieval. Existing works always separate the dual-encoder learning and ANN index building (Khattab and Zaharia, 2020; Zhan *et al.*, 2021b), which suffer from degraded retrieval performance. To address the problem, Zhang *et al.* (2021a) explored the joint training of the dual-encoder and the Product Quantization (PQ) (Jégou *et al.*, 2011) index. They introduced a trainable indexing layer, which is composed of space rotation, coarse quantization and product quantization operations. Later, Zhan *et al.* (2021a) proposed JPQ, which firstly utilizes K-Means to generate fixed discrete codes for documents and then only trains the query encoder and PQ Centroid Embeddings jointly. However, this method suffers from a degree of performance loss. Further, Zhan *et al.* (2022) proposed RepCONC, which is capacity to optimize index assignments of document embeddings with a constrained clustering process. Experimental results show the RepCONC achieves better retrieval effectiveness on two benchmarks.

Joint Learning with Re-ranker On the basis of the pipeline architecture, most existing works in the IR field focus only on one of components, independently of all the others. However, separating each component for IR systems building suffers from a few drawbacks and produces sub-optimal performance. In fact, apart from separately training each component (e.g., retrieval and re-ranking), it has shown that the retrieval and ranking tasks are related with each other (Huang *et al.*, 2020; Gao *et al.*, 2020a; Khattab and Zaharia, 2020). Based on these observations, Ren *et al.* (2021) proposed a joint training method for dense retrieval and re-ranking, where the relevance information can be transferred between the two components with a unified list-wise training approach. Different from this work, Zhang *et al.* (2021b) considered to jointly train the two components within an adversarial retriever-ranker (AR2) framework. Within the framework, the retriever aims to recall hard negatives to confuse the re-ranker, and the re-ranker learns to differentiate positives and hard negatives. In this way, the retriever and re-ranker can be enhanced iteratively.

Joint Learning with Reader 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.*, 2020b) combines a pre-trained dual-encoder (DPR (Karpukhin *et al.*, 2020)) as the retriever with a pre-trained Seq2Seq model (BART (Lewis *et al.*, 2020a)) as the generator for OpenQA tasks. The query encoder and the generator are fine-tuned end-to-end 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 built the EMDR^2 model, which initializes the dual-encoder retriever with BERT and builds the reader on top of T5. Compared with the stage-wise training, their method allows training signals to flow between the reader and the retriever. Experimental results demonstrate that their method achieves new state-of-the-art results on three benchmarks.

3.2.3 Generalization Ability

In many scenarios outside commercial web search, obtaining training labels is difficult and sometimes infeasible due to privacy constraints (e.g, the medical domain). Thus, the generalization ability of retrieval models is important in real-world scenarios. However, many PTMs-based retrieval models have been observed diminishing advantages over term-based retrieval models like BM25 in various benchmarks if they are not fine-tuned with adequate labels (i.e., the zero-shot setup). Specially, Thakur *et al.* (2021) studied whether the retriever models can generalize to other domains and concluded that the generalization ability of PTMs-based retrieval models is significantly worse than PTMs-based re-ranking models.

Some early works show great improvement under the zero-shot setting for dual encoders by leveraging strong training losses (Hofstätter *et al.*, 2021a) or synthetic data generation (Liang *et al.*, 2020; Ma *et al.*, 2021a; Reddy *et al.*, 2021). For example, TAS-B model (Hofstätter *et al.*, 2021a) with the training loss function based on knowledge distillation shows strong generalization capacity and better out-of-distribution performances. Ma *et al.* (2021a) proposed a data augmentation approach to leverage existing QA datasets to train a question generation model given the paired document. Then, the model can be applied to target-domain documents and generates queries for them. Then, these synthetic query-document pairs can be used to train a retrieval model. Recently, Ni *et al.* (2021a) challenged the belief in Thakur *et al.* (2021) that models with more interactions between queries and documents have better generalization ability. They explored the generalization ability of dual-encoder models by scaling up the model size while keeping the bottleneck embedding size fixed. Experimental results on the BEIR dataset (Thakur *et al.*, 2021) show that scaling up the model size brings significant improvement on a variety of retrieval tasks, especially for out-of-domain generalization. Besides, Xin *et al.* (2021) proposed MoDIR to improve the generalization ability of dense retrievers. Concretely, they introduced an auxiliary domain classifier into the dense retriever training to learn domain-invariant representations, where the retrieval model is not only optimized for the retrieval-orient objective, but also trained to confuse the domain classifier.

3.3 Summary

This chapter presents how pre-training methods are applied in the retrieval component.

Firstly, we review existing works within three basic model structures, including sparse retrieval models, dense retrieval models, and hybrid retrieval models. Sparse retrieval models employ pre-training methods to re-weight terms based on semantic features or map queries/documents into a latent word space to enhance term-based retrieval methods. Due to the sparsity of the representation obtained by sparse retrieval models, they can still utilize the existing inverted index for efficient retrieval. Dense retrieval models employ PTMs-based dual-encoder architecture to learn standalone low-dimensional dense representations for queries and documents, and then use approximate nearest neighbor search algorithms for fast retrieval. Equipped with pre-training methods, these models often show promising results and naturally obtain increasing research interests in this community. Hybrid retrieval models are composed of sparse retrieval models and dense retrieval models to absorb merits of both. As expected, these hybrid models usually show better retrieval performance, and at the cost, they require much higher retrieval complexity.

Secondly, we discuss several advanced topics of wide concern to researchers in this community, including negative sampling strategy, joint learning with other components, and generalization ability. For PTMs-based retrieval models, negative sampling is one of the most important elements for efficient and effective model learning. There have been extensive works focusing on exploring various negative mining methods. Moreover, the application of PTMs in the retrieval component makes the joint learning of other modules (e.g., index) or downstream tasks possible. Currently, there have been some preliminary works for this topic and it would be a promising direction for the future work. Although these PTMs-based retrival models have shown inspiring results on several popular benchmarks (e.g., MS MARCO and Natural Questions (Kwiatkowski *et al.*, 2019)), they are observed reduced advantages if are not fine-tuned with abundant task-specific labeled data. With the release of BEIR(Thakur *et al.*, 2021) benchmark, researchers begin to focus on improving the generalization ability of PTMs-based retrieval models. However, it is still in its infancy stage and worthy of further exploration.

4

Pre-training Methods Applied in the Re-ranking Component

In this section, we review previous works applying PTMs in the re-ranking component. After the efficient first-stage retriever, there can be a stack of complex re-rankers in the re-ranking stage where the input of each re-ranker comes from the previous one. Such a multi-stage cascaded architecture is commonly-used both in the industry (Yin *et al.*, 2016; Liu *et al.*, 2021g; Li and Xu, 2014) and the ranking leaderboard in the academia (Craswell *et al.*, 2021). Generally, PTMs are often employed to re-rank a small set of candidates provided from the first-stage retriever. By learning powerful representations or modeling complex interactions between queries and documents, PTMs have achieved great success compared with previous methods (Mikolov *et al.*, 2013a; Lin *et al.*, 2021a; Nogueira *et al.*, 2019b).

4.1 Basic Model Architecture

According to the two schools of relevance modeling, i.e., discriminative modeling or generative modeling, in the IR literature (Ponte and Croft, 2017; Robertson and Zaragoza, 2009), the methods applying PTMs in the re-ranking component can be categorized into three classes: 1) Discriminative Ranking Models: model $P(r, d|q)$ by directly learning a

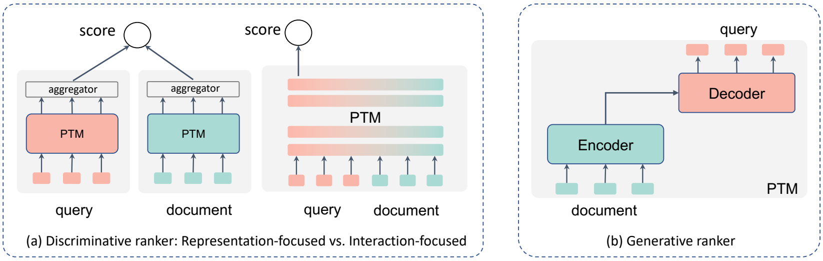


Figure 4.1: Two categories of re-ranker.

relevance “classifier” from labeled data; 2) Generative Ranking Models: approximate the true relevance distribution $P(r|q,d)$ by modeling the generative process between queries and documents; 3) Hybrid Retrieval Models: joint learn the discriminative model and generative model to leverage merits of both for better ranking performance.

4.1.1 Discriminative Ranking Models

From the very beginning (about 2015-2018), applying PTMs in the re-ranking component focused on leveraging the pre-trained word embedding such as word2vec (Mikolov *et al.*, 2013a) and GloVe (Pennington *et al.*, 2014) into discriminative ranking models (Guo *et al.*, 2016). These word embeddings are mainly used to initialize the embedding layer of ranking models, and other components are usually learned from scratch. Start with BERT, which pre-trains a Transformer model using self-supervised objectives on large-scale unlabeled corpora, both pre-trained word representations and interactions can be “transferred” to the ranking model. The former can be used in the same way as previous static word embeddings like word2vec, or like the latter that fine-tunes the whole pre-trained model and only a lightweight task-specific classification layer is learned from scratch. This is also known as the “pre-train and fine-tune” paradigm. It’s more convenient to fine-tune the whole model on downstream tasks as there is no need to design complicated model architectures for each task. BERT and its successors have achieved great success when applied in the re-ranking component in this way. This type of PTMs are generally pre-trained with self-supervised language modeling tasks, and the encoder is employed to build the

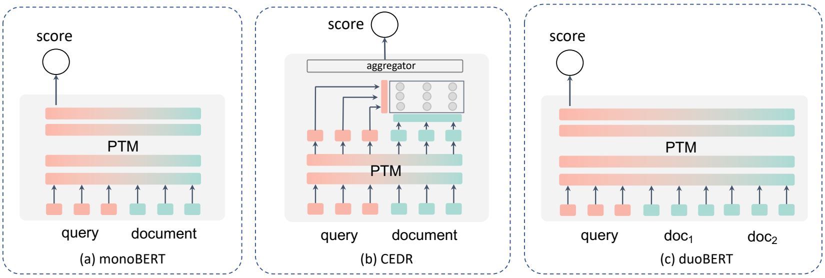


Figure 4.2: Three typical interaction-focused discriminative ranking models.

discriminative ranking model (Devlin *et al.*, 2019; Yang *et al.*, 2019). We term this type of PTMs as discriminative PTMs. Following the recipe of NeuIR (Guo *et al.*, 2020), there are also two ways in applying PTMs as the discriminative ranking model on the re-rank component, namely representation-focused models and interaction-focused models. We introduce them in detail in the following.

Representation-focused Models Representation-focused models (Figure 4.1 (a) left) usually adopt a bi-encoder architecture and encode queries and documents separately, and then the relevance score is computed with simple similarity functions between representations of queries and documents. Without loss of generality, the representation-focused method could be also abstracted by Eq. 3.1. ϕ_{PTM} and φ_{PTM} are PTMs which take the raw text of the query or the document as the input, and output one dense representation for each, respectively. ϕ_{PTM} and φ_{PTM} could share the parameters or not. Then, the relevance is computed by simple similarity functions f like cosine or MLP.

In the early days, representation-focused methods often employ pre-trained word embeddings to initialize the representation of input tokens, and the remaining parameters are all randomly initialized. For example, ARC-I (Hu *et al.*, 2014) trained 50-dimensional word embeddings on Wikipedia and Weibo data using the Word2Vec. The word embeddings are then fed into convolutional neural networks to obtain text sequence representations, and the relevance score is computed using a MLP based on the two text sequence representations. They found that fine-tuning the word embedding can further improve the performance compared with fixing them. More details about the word embedding based ranking model are referred to this survey (Mitra and Craswell, 2018).

More recently, the Transformer-based PTMs are introduced to fine-tune the entire model on downstream tasks, rather than just initializing the word embedding layer. For example, Qiao *et al.* (2019) proposed to utilize the BERT to encode the query and the document separately, and take the [CLS] embedding of the last layer as their representations and then calculate the ranking score via cosine similarity. Other studies have shown that using mean pooling on contextual embeddings of the whole input sequence performs better than the [CLS] embedding (Reimers and Gurevych, 2019). Qiao *et al.* (2019) have shown that representation-based architectures are less effective than interaction-based architectures, but they can be more efficient by utilizing approximate nearest neighbor (ANN) techniques to search from the pre-computed representations. Thus, the representation-based model architectures are usually applied to the first-stage retrieval phase (see Section 3.1.2).

In general, discriminative ranking models can be fine-tuned using the *pointwise*, *pairwise*, or *listwise* learning objectives following the learning to rank literature (Liu, 2007). However, the Transformer-based PTMs usually limit their input length to 512 due to the quadratic time and memory complexity of self-attention (Devlin *et al.*, 2019; Brown *et al.*, 2020). Therefore, long documents that contained more than 512 tokens will be truncated before being fed into the model, and more techniques about handling long documents for Transformer-based PTMs will be introduced in Section 4.2.1.

Interaction-focused Models Interaction-focused models (Figure 4.1 (a) right) aim to capture low-level interactions between terms in query-document pairs, and then calculate the relevance score based on their interaction features. For the usage of pre-trained word embeddings in interaction-focused models, they are also used to initialize the representation of input tokens as in representation-focused models (Mitra and Craswell, 2018). In this section, we mainly introduce how the Transformer-based PTMs are used as the interaction-focused model. Without loss of generality, the interaction-focused method could be abstracted as:

$$rel(q, d) = f(\eta_{PTM}(q, d)) \tag{4.1}$$

where η_{PTM} is the interaction function based on PTMs, and f is the scoring function that estimates the relevance score according to the interaction features. The input for η_{PTM} is a concatenation of the query and the document. In this way, the interaction of the query and the document could be modeled inside the η_{PTM} with the self-attention mechanism. Note that the interaction cannot be pre-calculated until the query comes, which implies that it's better to use these models re-rank a small set of documents due to the large cost of computing all query-document pairs in the collection.

The most immediate usage of pre-trained Transformers in the interaction-focused model is MonoBERT (Nogueira and Cho, 2019). It takes the concatenation of the query and the passage as inputs of the BERT, and feeds the [CLS] vector to a feed-forward network to obtain the relevance score. They take the *pointwise* loss function, i.e., the cross-entropy loss, to fine-tune the BERT model on the MS MARCO passage ranking task (Craswell *et al.*, 2021). It is interesting to see that such a direct use of BERT showed outstanding performances compared with previous NeuIR models. CEDR (MacAvaney *et al.*, 2019) stacks a traditional neural interaction model upon 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.*, 2016) and KNRM (Xiong *et al.*, 2017a). The [CLS] vector is also incorporated in CEDR to enhance the model's signals. CEDR is trained using pairwise hinge loss (Dehghani *et al.*, 2017b). By combining BERT and NeuIR models, CEDR is significantly better than the Vanilla BERT on Robust04 and WebTrack 2012–14. DuoBERT (Pradeep *et al.*, 2021) takes a sequence comprised of a query and two passages as input and is trained to estimate the positive candidate is more relevant than the negative. The advantage of DuoBERT is that it can explicitly model the document comparison for pairwise learning objectives. However, due to the length limitation of the BERT, the whole sequence is truncated to 512 tokens and each passage can have at most 223 tokens. Though its effectiveness as shown in the

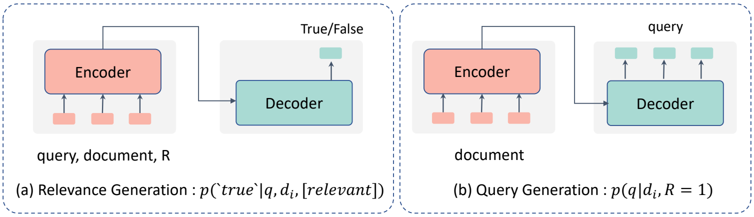

True/False
Encoder
Decoder
query, document, R
(a) Relevance Generation : $p(`true`|q, d_i, [relevant])$
query
Encoder
Decoder
document
(b) Query Generation : $p(q|d_i, R = 1)$

Figure 4.3: Two categories of generative ranking models.

passage ranking, the length restrictions largely hinders the application of duoBERT in document ranking tasks.

4.1.2 Generative Ranking Models

In addition to discriminative ranking models, researchers have also explored the usage of PTMs as generative ranking models (Roy *et al.*, 2016b; Santos *et al.*, 2020; Nogueira *et al.*, 2020). The generative ranking model, which aims to approximate the true relevance distribution, has studied a lot in last century such as statistical language models (Ponte and Croft, 2017), and classic probabilistic relevance models (Robertson and Jones, 1976). Statistical language models like 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, 2017; Zhai, 2007). Classic probabilistic relevance models like Binary Independence Model focuse on describing how a document is generated from a given query (Lafferty and Zhai, 2003; Robertson and Jones, 1976).

For word embedding-based PTMs, they can be easily incorporated into statistical generative retrieval models to compute the semantic similarity between terms (Ganguly *et al.*, 2015; Zuccon *et al.*, 2015). For example, word embeddings can be used to augment the language modeling (Zamani and Croft, 2016) or the translation modeling (Zuccon *et al.*, 2015) in the generative ranking model by computing the semantic similarity between terms. For Transformer-based PTMs, they pre-train the decoder of the Transformer or the whole Transformer (i.e., encoder-decoder) with autoregressive language modeling tasks like causal language modeling (Brown *et al.*, 2020). Then, the pre-trained generative model can be applied to either generate the query or the relevance label. We term this type of PTMs as generative PTMs. Recent works on applying generative PTMs to re-ranking are mainly based on the 1) Query Generation process, which is inspired by the query likelihood model. Another line of researches studied the 2) Relevance Generation process which generates a specified relevance token given the query and the document.

Query Generation The first type of generative ranking models is based on the query generation process. The basic idea is to rank documents by the likelihood of generating the query from documents using generative PTMs like GPT (Brown *et al.*, 2020) and BART (Lewis *et al.*, 2020a). Without loss of generality, the query generative models could be abstracted as

$$rel(q,d) = f(\phi_{PTM}(q|d)) = \prod_{i=1}^{|q|} \phi_{PTM}(q_i|d), \tag{4.2}$$

where ϕ_{PTM} is the generative PTMs and f is a multiplication function $\prod$. Given the document d, each query term q_i is generated one by one and the relevance score is thus obtained by multiplying their normalized probabilities. The usual approach to train such generative models is to use maximum likelihood estimation (MLE). Note that, at inference time, the model also uses the Teacher Forcing strategy like the training process. That is, for each generation, the oracle query term (i.e., ground truth) is used as input for generating the next, instead of model output from a prior time step.

A direct usage of pre-training methods in query generation is to take generative PTMs like GPT and BART to estimate the probability in generating queries. Santos *et al.* (2020) proposed a query generative model for ranking answer passages in QA. They take the conditional likelihood of generating a question against a passage as the relevance score following Eq. 4.2. Two types of loss function is proposed to take advantage of both the positive and negative examples: 1) likelihood and unlikelihood loss (LUL) based on MLE; 2) a pairwise ranking loss (RLL) such as a margin loss based on their likelihood. Experiments results showed that RLL loss is very helpful for training query generative ranking models. In addition, they also observed that the generative ranking models can generate fluent questions. Finally, they found that the query generative models are as effective as simple discriminative ranking models for answer selection.

Relevance Generation Relevance generation is focused on generating specified relevance tokens by feeding the concatenation of the document and the query into the generative PTMs, and the probabilities of these relevance labels are treated as relevance scores. Without loss of generality, the relevance generative models could be reformulated as:

$$rel(q, d) = f(\phi_{PTM}(t|q, d)), \tag{4.3}$$

where t is the relevance tokens. In essence, the relevance generation is a classification task as the model is trained using pointwise loss function on relevance tokens and ranks documents by the probability of predicting the target relevance token.

Considering the relevance token generation is more like a classification task, it can be modeled by both generative PTMs and discriminative PTMs. Nogueira *et al.* (2020) proposed to use the generative PTMs T5 for modeling relevance generation. As T5 is a unified text-to-text language models, they also devised a text-to-text template for the ranking task where the input is "Query: [q] Document: [d] Relevant:" and the output is "true" or "false". T5 is fine-tuned to generate the target tokens instead of directly producing relevance probabilities. The probability of the "true" token is used to represent the document relevance score, which is normalized with softmax function over the logits of "true" and "false" tokens . Other target tokens like "yes/no" perform worse than the "true/false" tokens. Experiments show T5-3B, which was firstly trained on MS MARCO passage ranking task, outperforms some supervised training models like Birch, BERT-maxP and PARADE, in a zero-shot manner on Robust04.

Moreover, the above method is similar to the prompt learning where the model is guided to predict the "label" based on prompts (Schick and Schütze, 2021a; Schick and Schütze, 2021b). A template and a verbalizer are needed to design first for a given task, where the template is used to transform the original text to a specific form, and the verbalizer is used to project original labels to some words which are fit for the template. Take a sentiment classification task as an example, assume the template is "[text] It is [mask]" in which the token [text] represents the original text, and the token [mask] stands for the verbalized words such as "great" and "terrible". These two words are mapped from the positive label and the negative label, respectively. The PTMs are trained to predict the probability distribution on the [mask] position given the text with a specific form. On some NLP tasks, the prompt learning has shown exciting results under the few-shot setting. It might be that the reformatted task is almost identical to MLM, which makes it a better usage of pre-trained knowledge (Lester *et al.*, 2021; Li and Liang, 2021). However, how to leverage the prompt learning to improve the few-shot learning in IR has not been explored at this point.

4.1.3 Hybrid models

Combining the generative and the discriminative modeling leads to the hybrid models. Liu *et al.* (2021a) proposed a multi-task learning approach to jointly learn the discriminative and the generative relevance modeling in a unified pre-trained model. They assumed that joint these two different types of retrieval modeling leads to better generalized, and hence more effective retrieval model. To verify this hypothesis, they leveraged the generative PTMs (i.e., BART) or the discriminative PTMs (i.e., BERT) to learn discriminative ranking tasks as well as other language generation tasks, such as query generation task, questions generation task, and anchor text generation task. For the generative PTMs, they fed the document and the query into the encoder and the decoder respectively. Then, the query is generated in a sequence-to-sequence manner and the relevance score is calculated by the last token of the entire sequence using a feedforward layer. Since the bidirectional attentions in BERT cannot fully adapt to the sequence-to-sequence training strategy, they implemented a mix of attention mechanisms including bidirectional attention, unidirectional attention and cross attention to support sequence-to-sequence tasks. Their experiments showed that jointly learning discriminative tasks and generative tasks leads to significant improvement on the MS MARCO passage ranking task.

4.2 Advanced Topics

In addition to the direct application of PTMs in IR, researchers have also developed a considerable amount of studies to address the IR-specific challenges. On one hand, the document length varies significantly across different domains, where PTMs often fail to address the long document due to the length restriction of the input. On the other hand, PTMs often consist of a large number of parameters which would increase the search latency. In what follows, we will introduce researches in addressing these two problems.

4.2.1 Long Document Processing Techniques

In the traditional ad-hoc retrieval, documents always contain thousand of tokens in standard TREC datasets (Voorhees, 2004; Dietz *et al.*, 2017). However, due to the quadratic time and memory complexity of self-attention mechanism in modern Transformer-based (Vaswani *et al.*, 2017) PTMs, the length limit of input is always up to 512. A majority of applications are to segment the long document text into smaller chunks that can be processed by the PTMs and then do an aggregation over chunks. Based on the aggregation type, these methods can be broadly categorized into two classes: 1) Passage Score Aggregation: aggregate the relevance score of the query and segmented passage; and 2) Passage Representation Aggregation: aggregate the representations of segmented passages to document representations first and then compute the relevance between query and the aggregated document representations.

Passage Score Aggregation Passage score aggregation is a postprocessing method that only aggregates the relevance score between the query and the segmented passages provided by the PTMs. Different methods focus on designing document segmenting and aggregate function (Dai and Callan, 2019b; Hofstätter *et al.*, 2021b).

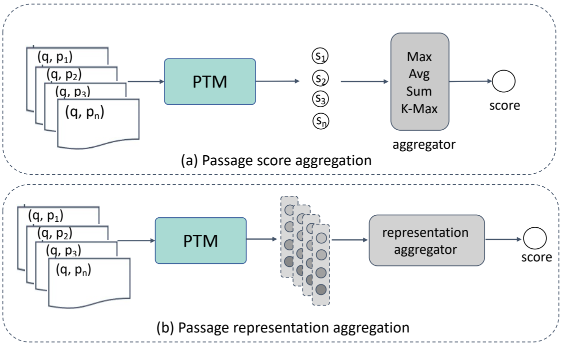


Figure 4.4: Two categories of passage aggregation methods.

Dai and Callan (2019b) proposed to split a document into overlapping passages using a 150-word sliding window. Each passage will be concatenated with the query to input to the BERT, and the relevance of each passage is predicted independently. Three methods are proposed to aggregate the relevance scores of passages: 1) BERT-firstP that only uses the score of the first passage; 2) BERT-maxP that uses the maximum score of the passages; 3) BERT-sumP that sums all the relevance scores of passages. 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 vice versa. However, according to the *Scope Hypothesis* (Robertson and Walker, 1994), the document could be partially relevant to a query and thus not all passages are relevant to a query. There will be noise in the data if we treat all the passages as positive to the query. BERT-maxP and BERT-sumP perform better than BERT-firstP on traditional ad-hoc retrieval tasks including Robust04 and ClueWeb09-B in their experiments since all passages are taken into account. But these two methods require more computational cost as all the query-passage pairs need to be trained and predicted while BERT-firstP only considers the first passage of each document. IDCM (Hofstätter *et al.*, 2021b) divides the document into multiple fixed-size windows of 64 words with overlapping of 7 words for both the previous and latter, respectively. The basic idea is to firstly take 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. Since some operations in the IDCM framework are not differentiable like passage selection module, therefore, they adopts a three-stage optimization pipeline to training the model. Specifically, ETM and ESM are trained separately where ETM is first optimized on passages and then on full documents by aggregating the score of top passages, ESM is distilled from ETM. IDCM achieves comparable effectiveness to the BERT-based ranker on two benchmarks including TREC DL 2019 and MS MARCO passage ranking, but with lower computation cost and query latency. The main restriction is that this method is a little bit complicated on the model framework and the training process. Although aggregating passage score is simple and effective, it loses the long-range dependence over the whole document as it uses one passage to estimate the relevance independently every time.

Passage Representation Aggregation Instead of only aggregating the passage score, aggregating the passage representation seems more convincing in which the relevance score is estimated by considering all the passages together.

PARADE (Li *et al.*, 2020a) segments the long document into a fixed number of overlapping chunks using 225-word sliding windows. Then all passage representations from a document are aggregated for estimating the document relevance score. They proposed two types of passage aggregation method: using a mathematical operation such as the elementwise mean, max and sum on the representation vectors; or using a deep neural network including MLP, convolutional neural networks and Transformer layers. By aggregating the representations with more complicated architectures, PARADE$_{Transformer}$ can significantly improve the performance over passage score aggregation methods like BERT-maxP and other passage representation aggregation methods like PARADE$_{max}$. 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 first studied the accumulation process of patterns of the passage-level information from a user's information seeking perspective. They show the sequence of passage-level cumulative gain can be effectively predicted as a sequence prediction task. Then, BERT is employed to learn representations of each query-passage pair and then a LSTM network is adopted to aggregate passage representations and predict the passage cumulative gain. The cumulative gain of the last passage is treated as the document-level gain or the document-level relevance score. The model is trained on graded passage-level relevance judgments to predict the cumulative gain of previous passages. Experiments on two Chinese datasets show its effectiveness in improving ranking performance. The main limitation is that labeling the passage-level relevance judgments is too expensive.

4.2.2 Model Acceleration

Efficiency is one of the major concerns for applying PTMs in IR as there is always a large-scale data in the real search scenario. Since the Transformer-based PTMs often consist of tremendous amount of parameters ranging from millions to billions, this greatly increase the computational cost and memory storage. So it's hard to deploy these PTMs on the online service in real-world applications or on resource-restricted devices considering their requirement of low latency. To address this issue, researches have explored several methods to reduce the high computational cost in the re-ranking stage including decoupling the interaction of the query and the document, model distillation, dynamic modeling, and lightweight fine-tuning.

Decouple the Interaction One of the bottlenecks that limits the efficiency of the Transformer-based PTMs comes from the self-attention mechanism. In the re-ranking stage, the interaction-focused ranking models that apply Transformer-based PTMs are widely-used and more effective than representation-focused ranking models. But the representation-focused models are more efficient as they can pre-compute the document representations to reduce the online inference time. Researchers have studied to incorporate the advantage of the representation-focused architectures into the interaction-focused architectures.

PreTTR (MacAvaney *et al.*, 2020) employed the BERT model and proposed to decouple the low-level interaction of the query and the document via encoding them separately and then interacting in the late BERT layers. Thus, the document representations can be pre-computed offline and only the query needs to be encoded online. So most computational budget comes from the interaction of the last few layers now. When merging the query representation and document representation on layer 11 of BERT, PreTTR achieved a 42X speedup on TREC Web-Track while not significantly reducing the ranking performance. But merging them at layer 11 performs poorly on the Robust04 dataset. This indicates that merging the representation of queries and documents at which layer depends on the datasets. When the query and the document encoding is totally decoupled, it degrades to the representation-focused architecture. Thus, it's a trade-off between efficiency and effectiveness. MORES (Gao *et al.*, 2020a) proposed a similar idea to improve the efficiency of the BERT-based re-ranker in which they modularize the Transformer-based neural re-ranker into two separate modules, i.e., text representation module and interaction module. One of the main differences is that the interaction module in MORES is not a fully cross-attention mechanism. It canceled the document-to-query attention, and only query-to-document attention is performed followed by query self-attention. In this way, the document representation is kept unchanged for all queries. Experiments on MS MARCO passage ranking and TREC 2019 passage ranking showed that 2 layers of lightweight interaction module can achieve ranking performance competitive with a fully interaction-focused architecture while achieving tens or hundred of speedup.

Model Distillation Knowledge distillation is a widely used method for reducing the computational cost by transferring knowledge from the teacher to the student. The basic idea is to learn a smaller model from the outputs of a larger teacher model (Hinton *et al.*, 2015; Sanh *et al.*, 2019). Hinton *et al.* (2015) proposed a simple yet effective method that transfers the final logits of the teacher on labeled data and unlabeled data to the student where the teacher is first trained on supervised data.

Other studies also investigate to transfer the intermediate hidden states or the attention matrix (Jiao *et al.*, 2020). For the Transformer-based PTMs, there are many studies to verify its effectiveness on various tasks including in IR.

Gao *et al.* (2020b) investigated three methods to distill BERT for ranking, including only distilling the ranking information of the search task (Ranker Distill), distilling the MLM information over a large text corpus followed by a normal fine-tuning on the search task (LM Distill+Fine-tuning), and distilling both (LM Distill+Ranker Distill). The teacher model uses BERT-base which contains 12 layers of Transformer, and the student model uses a 4 or 6 layers of Transformer. 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. The 6-layer distilled BERT ranker(2X speedup) using the LM Distill+Fine-tuning method is able to achieve comparable performance to the original BERT, while the performance of the 4-layer distillation BERT ranker (9X speedup) drops significantly. On top of the TinyBERT model (Jiao *et al.*, 2020), Chen *et al.* (2021c) explored to distill the student model with three other kinds of internal weights of the teacher model simultaneously only 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 also benefit the ranking.

Dynamic Modeling Dynamic modeling which can adapt the model structures or parameters to different inputs is another promising method that can improve the efficiency of big models (Han *et al.*, 2021b). Dynamic modeling can selectively activate some model components of the whole model, such as some layers or a sub-network, conditioned on different inputs, and thus allocate computations on demand at the inference stage. For example, easy samples will have less computation as they can be predicted quickly with a high confidence. Early exit is a representative method in this line of research, which allows the examples to exit at early layers of the model without passing through the entire model.

It is natural to apply the idea of early exit to PTMs on ranking tasks, since most irrelevant documents can be easily predicted given the query. Xin *et al.* (2020a) employed such idea from 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 classifier of the *ith* layer is confident about the prediction of the sample, early exiting is performed and subsequent transformer layers are skipped. Note that the positive confidence threshold and the negative threshold in their paper are set to different values as they assume that positive (relevant) documents need more computations and the confidence score of positive documents is not only the exiting criterion but also the score for re-ranking. Experiments on the MS MARCO passage ranking dataset showed early exiting is able to accelerate inference by about 2.5X while maintaining the effectiveness of the original model. Cascade Transformer (Soldaini and Moschitti, 2020) is a sequence of re-rankers built on top of RoBERTa, that is, each re-ranker is a sub-network contained several Transformer layers and a new classification layer, one after another. When a batch is fed into the Cascade Transformer, each re-ranker will prune a subset of candidates and input the rest to the next until meet the last re-ranker. In this way, only a small set of candidates in one batch is passed through the whole model and most are pruned early. To enable this approach, the parameters of all re-rankers are trained in a multi-task learning fashion, in which one of the re-rankers is sampled to train and update the layers below the selected re-ranker for every mini-batch. Experiments showed that the Cascade Transformer can get competitive performance to the original RoBERTa while largely reducing the computational cost (over 37% per batch).

Lightweight Fine-tuning The most common way to apply PTMs is to fine-tune all the parameters given the data from the downstream task. For the word embeddings, they can be fixed alone or fine-tuned along with the whole neural model without adding too much computation. However, for the Transformer-based PTMs, fine-tuning the whole model parameters often requires large computation costs and also storage spaces, especially when serving a large number of tasks with different big models. With the ever-increasing size of Transformer-based PTMs, ranging from millions (Devlin *et al.*, 2019; Brown *et al.*, 2020) to billions (Brown *et al.*, 2020) or even trillions of parameters (Fedus *et al.*, 2021), fully fine-tuning gradually became impossible for a regular community. To mitigate this issue, researchers investigate several lightweight fine-tuning strategies that updates only a small number of extra parameters of PTMs while keeping most pre-trained parameters frozen. In this way, we can not only reduce the computation cost to improve the efficiency but also store only one big model and many tunable extra parameters for various tasks.

The intuitive method of lightweight fine-tuning is to freeze some or all pre-trained parameters, but this will hurt the performance greatly without some specific designs (Houlsby *et al.*, 2019). Another line of research studied to insert small neural modules into existing models and only these inserted modules are fine-tuned on the downstream task. For example, Houlsby *et al.* (2019) proposed to insert adapters at each layer, which is a MLP with a non-linear function that projects the input vectors down first and then up. Li and Liang (2021) proposed prefix tuning that prepends several additional prefix tokens to the input or hidden layers, and only these prefix tokens are fine-tuned on downstream tasks. Hu *et al.* (2021) proposed LoRA that learns low-rank matrices for the attention matrix to approximate parameter updates. Researchers have also explored these methods into IR tasks. Jung *et al.* (2021) examined the above lightweight fine-tuning methods in the PTMs-based ranking models. They used a BERT-based bi-encoder architecture for the re-ranking stage. Experiments on three standard ad-hoc retrieval tasks, including Robust04, ClueWeb09-B and MS MARCO document ranking dataset, showed the effectiveness of these lightweight fine-tuning methods. In addition, they also proposed a semi-Siamese bi-encoder architecture to reflect the different characteristics of query and document based on the lightweight fine-tuning methods. For example, when applying prefix-tuning, they add different prefixes for query encoder and document encoder besides a common prefix. Experiments also demonstrate that such a design can enhance the ranking performance on these datasets.

4.3 Summary

In this chapter, we first review the basic usage of PTMs when applying in the re-ranking component. According to the two schools of relevance modeling in the IR literature, we categorize these works into three classes, i.e., discriminative ranking models, generative ranking models and hybrid ranking models. 1) The word embedding methods are either used to initialize the embedding layer of discriminative neural ranking models or incorporated into the traditional statistical generative models. But the recent PTMs pre-train a very deep Transformer model and then fine-tune the whole model on downstream tasks which is proven to be more convenient and powerful. 2) The discriminative ranking models with PTMs can be modeled with representation-focused architecture or interaction-focused architecture. The representation-focused architecture is more efficient since it can pre-compute the document representations and only the query is encoded online. The interaction-focused architecture is more effective but with more computational costs as it needs to encode every query-document pair. 3) The generative ranking models with PTMs considered two kinds of generation processes, including the query generation and relevance token generation. The document generation hasn't been studied due to the difficulty of generating long texts conditioned on short texts. Inspired by the model-based IR system (Metzler *et al.*, 2021), the model may directly generate the document identifier given the short query instead of the whole document text. 4) The hybrid ranking models jointly learn discriminative ranking objective and query generation using multi-task learning. Existing approach does not show too much superiority and requires further exploration. Compared with the generative ranking models, the interaction-focused discriminative ranking models achieved better results on the re-ranking stage (Santos *et al.*, 2020). But the document identifier generation is also worthy of further exploration considering its efficiency and no needs to store documents.

We then introduced some advanced topics on applying PTMs in the re-ranking component, such as the long document processing techniques and various strategies to improve its efficiency. 1) Since the quadratic time and memory complexity of self-attention mechanism in the Transformer, most Transformer-based PTMs limit the input length up to 512 which is often not enough for web documents. Researchers have studied two approaches to handle long documents including passage score aggregation and passage representation aggregation, and the former is easy to use while the latter performers better (Li *et al.*, 2020a). 2) Although only a small set of documents are re-ranked, efficiency is also one of the major concerns of applying PTMs, especially the deep Transformer-based PTMs. Recent studies mainly focused on decoupling the interaction of the query and the document for the interaction-focused models, model distillation, dynamic modeling and lightweight fine-tuning. But all existing works have made a compromise, such as increasing training budget (e.g., model distillation and lightweight fine-tuning) or at the expense of performance (e.g., decoupling the interaction and dynamic modeling). In the future, model quantization an pruning (Ganesh *et al.*, 2020) may be worth trying as they can reduce both the model size and the training cost without losing (too much) performance.

5

Pre-training Methods Applied in Other Components

In this section, we review existing works in applying PTMs in other components of a search system, such as query expansion, query rewriting, document summarization, snippet generation, etc. To elaborate, we divide these works into three categories: I) Query Processing, II) User Intent Understanding, and III) Document Summarization. In the next, we will introduce the pre-training methods applied in these components, respectively.

5.1 Query Processing

To better bridge the gap between query text and document text, search systems usually contain a query processing module to rephrase the input queries. Generally, corresponding tasks include query expansion and query rewriting.

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. Earlier, a large body of work aimed at expanding the original query with the pre-trained word embeddings (Kuzi *et al.*, 2016; Roy *et al.*, 2016b; Diaz *et al.*, 2016; Zamani and Croft, 2016). For example, Zamani and Croft (2016) proposed to use word embeddings to incorporate and weight terms that are semantically similar to the query terms and further described two query expansion models which are based on embeddings. Similarly, Kuzi *et al.* (2016) leveraged the terms to expand the original query or incorporate them with the effective pseudo feedback-based relevance model.

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 grammar structures and concepts to 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 from the top-ranked documents to expand queries, 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-tail queries or questions into popular or frequent ones, 2) reformulate ambiguous input queries into well-formed queries to improve retrieval performance. In the pre-BERT age, some researchers proposed non-contextualized embedding-based approaches for query rewriting (Grbovic *et al.*, 2015; Grbovic *et al.*, 2016). By jointly modeling query content and the corresponding context within a search session, Grbovic *et al.* (2015) propose a novel rewriting method based on a query embedding algorithm. Their approach maps queries into vectors which are close in the embedding space to allow query expansion via simple K-nearest neighbor search.

To enhance conversational search, Lin *et al.* (2020) utilized traditional IR query reformulation techniques to realize historical query expansion (HQE) and then applied the T5-base (Raffel *et al.*, 2020) model for neural transfer reformulation (NTR), i.e., rewriting a raw utterance into a natural language question without coreference and omission. 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).

5.2 User Intent Understanding

In complex search scenarios, users may interact with the search system for multiple rounds. During this process, search systems should understand users' evolving intent to better satisfy their information needs. Besides modeling users' short-term intent with historical signals, the system can also forwardly provide assistance for search users. Related tasks include query suggestion, search clarification, and personalized search.

5.2.1 Query Suggestion

As users' search intents become complex nowadays, a single query usually cannot 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.*, 2017a), 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.

5.2.2 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, Habibi *et al.* (2016) utilized low-dimensional word embeddings learned by word2vec to clarify questions asked by users during a meeting. From another point, 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 via crowdsourcing based on the faceted or ambiguous topics in the TREC Web track. Later, Hashemi *et al.* (2020) introduced Guided Transformer (GT), which utilizes external information such as the 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 next clarifying question selection and document retrieval 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 (Carbinell and Goldstein, 2017) 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.

5.2.3 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. Some researchers have attempted to use word embeddings to enhance the personalized search (Kuzi *et al.*, 2017; Amer *et al.*, 2016). For example, Amer *et al.* (2016) realized the personalized query expansion with the word embeddings learned on the user's profile. Their work concluded that personalized word embeddings fail to improve the ranking results. However, Kuzi *et al.* (2017) found that using personalized word embeddings can slightly improve the performance of E-mail search.

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.*, 2020; 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 optimize 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.

5.3 Document Summarization

As most documents contain complicated information, it may take search users a long time to carefully comprehend the whole document. For users' convenience, modern search engines usually provide a specific piece of text as the preview for a landing page, a.k.a., search snippet. In some domains, keywords can also be given to enhance the search and classification of the corpus.

5.3.1 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 and Lapata, 2019; Zhong *et al.*, 2020; Wang *et al.*, 2019; Xu *et al.*, 2020; Zhong *et al.*, 2019) and abstractive summarization (Zhang *et al.*, 2020a; Dou *et al.*, 2021; Lewis *et al.*, 2020a; 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. Besides CNN, pre-trained word embeddings have also been adopted for document summarization (Kobayashi *et al.*, 2015; Kågebäck *et al.*, 2014; Mohd *et al.*, 2020). Generally, they aggregated the word embeddings within a document to represent the whole document and then calculated the semantic similarity at document-level to extract a summary.

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.*, 2013a) 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 and Lapata (2019) proposed a general framework called BERTSUM [1] 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 the evaluation metrics and the training objective by merely optimizing sentence-level ROUGE, Bae *et al.* (2019) presented a novel training approach that directly maximizes summary-level ROUGE scores through reinforcement learning (RL). Their method can achieve better performance in the abstractive summarization task. To combine auto-encoding with partially auto-regressive language modeling tasks, Bao *et al.* (2020) took Transformer as the backbone network to pre-train a unified language model UniLMv2. They designed a novel training procedure to jointly pre-train a bidirectional

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

language model (LM) for language understanding and a sequence-to-sequence LM for language generation, namely pseudo-masked language model (PMLM). Based on this technique, UniLMv2 performs better than other base-size pre-trained models such as BERTSUMABS and MASS in fine-tuning (Song *et al.*, 2019).

While most approaches only involve pre-training tasks such as token or sentence masking, BART (Lewis *et al.*, 2020a) corrupts raw text with more noising functions (such as token deletion, sentence permutation, text infilling, 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 bypasses the difficulty of summary-level optimization based on contrastive learning by taking extractive summarization as a semantically text matching problem. 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.*, 2020a), which adopts the gap-sentence generation (GSG) task tailored for abstractive summarization while pre-training. They hypothesized that exploiting a pre-training objective that is more similar to the downstream task may lead to faster and better performance when fine-tuned. 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 abstractive 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.*, 2021), on II) distilling large pre-trained Transformers for summarization (Shleifer and Rush, 2020), or on III) legal domain related tasks (Huang *et al.*, 2021).

5.3.2 Snippet Generation

Different from generic document summarization, search snippets should highlight relevant points 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.*, 2020a; 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 §5.3.1) 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 concatenated the query with the document and then fed them into the encoder to incorporate the query relevance. 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 BERT-based query-focused summarization model. 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 may be 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.*, 2020a; Savery *et al.*, 2020) are more suitable for Question-Answering (QA) system as they combine reading comprehension with language modeling.

5.3.3 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 the existing works 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.*, 2021d). There exists a large body of research aiming at leveraging pre-trained word vectors for keyphrase extraction (Wang *et al.*, 2014; Qiu *et al.*, 2019; Papagiannopoulou and Tsoumakas, 2018; Mahata *et al.*, 2018). For instance, Wang *et al.* (2014) proposed a graph-based ranking approach that uses information supplied by word embedding vectors as the background knowledge. They further performed keyphrase extraction by constructing a weighted undirected graph for a document to compute the final scores of words.

From another angle, 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 an attention layer to automatically extract keywords from 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 approaches 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 another line of research on designing PTMs tailored for IR (Zamani and Croft, 2017; Lee *et al.*, 2019b; Chang *et al.*, 2020; Ma *et al.*, 2021b; Ma *et al.*, 2021d; Gao and Callan, 2021a; Chen *et al.*, 2022). Initially, PTMs were designed for NLP and the goal is to learn good representations for words or texts. When applying original PTMs in IR, studies have demonstrated that they can also benefit many IR tasks, since it's one of the basic requirements for IR to build good representations for queries and documents. However, the core of IR is to model the notion of **relevance** (Lavrenko and Croft, 2017; Saracevic, 2016; Fan *et al.*, 2021), which is not considered in the existing PTMs designed for NLP. To address this issue, researchers in the IR community have also started rethinking and exploring new pre-training objectives as well as architectures from the IR perspectives.

Without loss of generality, the general ranking function could be further abstracted as

$$rel(q, d) = f(\phi(q), \psi(d), \eta(q, d)), \tag{6.1}$$

where ϕ and ψ are representation functions to extract representation features, η is the interaction function to extract interaction features, and f is the scoring function which is usually a simple function like cosine or a MLP. According to the role of the PTMs in the ranking function, we divide them into two categories: 1) Pre-training Embeddings/Representation Models for IR; 2) Pre-training Interaction Models for IR.

For example, traditional word embedding methods take a single text sequence as input and output a fix-dimensional vector for each word. So the output word embeddings are usually employed to model the representation functions ϕ, ψ. The recent Transformer-based PTMs have two kinds of pre-training methods based on the input format and the pre-training objectives. The first one takes a single text sequence as input and learns contextualized word representations with various language modeling tasks (Liu *et al.*, 2019; Yang *et al.*, 2019), and this type of PTMs can be categorized into pre-trained representation models to model ϕ, ψ. The other one takes a text sequence pair as input to directly learns their interactions (Devlin *et al.*, 2019; Wang *et al.*, 2020; Lan *et al.*, 2020), and this type of PTMs can be categorized into pre-trained interaction models. Note that the pre-trained representation models can also be applied to the interaction-focused architecture by fine-tuning on labeled data, and vice versa. However, this will create the pretrain-finetune discrepancy which may not activate their full power of pre-training.

6.1 Pre-training Embeddings/Representation Models for IR

Pre-trained word embeddings (Mikolov *et al.*, 2013b; Pennington *et al.*, 2014) are mainly used to initialize the word embedding layer of a neural ranking model while pre-trained representation models (Liu *et al.*, 2021f; Brown *et al.*, 2020; Liu *et al.*, 2019) can be fully "transferred" to IR tasks without designing additional model architectures. That is, we can fine-tune the entire pre-trained representation model with supervised data on downstream tasks. Fine-tuning pre-trained models has become the de facto learning paradigm in many fields including NLP and CV. Methods in this category including static word embeddings and representation models are all pre-trained with self-supervised tasks on large-scale corpora. We introduce them in detail next.

6.1.1 Static Word Embeddings

Typical static embedding methods designed for NLP are trained based on word co-occurrence, especially the word proximity, in a large corpus. By predicting the adjacent word (words) given the context words (word) occurring within a local window, they can capture some lexical, syntactic, and semantic features of words. Although these word embedding methods have been widely used in neural ranking models and demonstrated to be effective in a number of IR tasks, they are not necessarily equivalent to the primary objective of IR. The main objective of IR is to predict the words observed in the documents relevant to a particular information need (Zamani and Croft, 2017). Previous studies investigated to design word embedding methods tailor for IR mainly from two aspects: 1) Regularizing the Original Loss towards IR characteristics; 2) Designing New Objectives to capture relevance. We only briefly describe some representative methods in these two lines of research.

Regularizing the Original Loss Some IR-specific characteristics are not considered in the typical word embeddings designed for NLP, such as document-level word frequency and text length, adding these clues to the learning objectives can further improve its effectiveness on IR tasks. Ai *et al.* (2016b) found the original paragraph vector (PV) (Le and Mikolov, 2014) 1) could suppress the importance of frequent words in a document excessively, 2) prone to over-fit short documents during the training process, and 3) ignores to model word-context associations in the learning objective. Thus, they proposed three modifications to regularize the existing loss function including idf-based negative sampling, introducing L2 to regularize document length, and adding another objective for learning paradigmatic relations.

Designing New Objectives Besides regularizing the original loss functions, researchers also explored to design new learning objectives for word embeddings. Diaz *et al.* (2016) proposed to train local word embeddings in a query-specific manner, that is, using query and the top-k documents retrieved by a statistical language model approach (Croft and Lafferty, 2003) to capture the nuances of topic-specific language. But, this model needs to be trained during the query time and thus is not always practical in real-word applications. Zamani and Croft (2017) pre-trained unsupervised relevance-based word embeddings by predicting the words that occurred in the top-k retrieved documents given the query words under the word2vec framework. The difference is that they use pseudo-relevance feedback (PRF) models, especially the relevance based language model (Lavrenko and Croft, 2017), to retrieve documents offline. They used a very shallow neural network which is a feed-forward neural network with a single linear hidden layer, to train the relevance word embeddings on millions of queries. Experiments on query expansion task and query classification task showed that the expansion terms chosen by their models are more related to the whole query than word2vec. Gysel *et al.* (2018) proposed another unsupervised model for document retrieval, called NVSM, in which the hypothesis of the optimization objective is that word sequences (i.e., n-grams) extracted from a document should be predictive of that document. Specifically, multiple phrases of n contiguous words are sampled from a document and then train the averaged word representations of phrases to predict the corresponding document representations. Experiments show that NVSM outperforms other latent vector space models like word2vec. Encouraging the n-grams and the document to be close may introduce noise as the randomly sampled n-grams may semantically similar to many documents.

6.1.2 Representation Models

Static word embeddings cannot model polysemy as the use of these words varies across linguistic contexts. To address this issue, previous methods also proposed to learn context-dependent representations (Melamud *et al.*, 2016; McCann *et al.*, 2017; Peters *et al.*, 2018). With the development of representation learning, researchers have studied pre-training a whole deep neural model like Transformer (Vaswani *et al.*, 2017) with self-supervised tasks for the contextualized word representations, and then transferring the entire model to the downstream tasks (Liu *et al.*, 2021f; Brown *et al.*, 2020; Liu *et al.*, 2019). The self-supervised tasks are mainly language modeling tasks, such as causal language modeling, masked language modeling and permuted language modeling (Qiu *et al.*, 2020). Although these PTMs can produce good contextualized word representations, studies have shown that they yield rather bad text sequence embeddings, often worse than averaging GloVe embeddings (Reimers and Gurevych, 2019). Hence, researchers investigate to pre-train high-quality text sequence representations for queries and documents. And the pre-trained representation models are often employed in the representation-focused ranking models. These works on pre-training representation models for IR are mainly from two aspects: 1) Pre-training Objectives; 2) Model Architectures.

Pre-training Objectives According to the underlying hypothesis of learning objectives, previous works can be categorized into two classes. The first assumes that if the pre-training objective resembles the downstream task, PTMs can achieve faster and better performance in the fine-tuning stage. Lee *et al.* (2019b) proposed a new pre-training task for passage retrieval in open domain question answering (openQA), i.e., Inverse Cloze Task (ICT), where one sentence is randomly sampled from a given passage as pseudo query and the rest sentences are treated as its positive context. Inspired by ICT, Chang *et al.* (2020) proposed another two tasks to better take advantage of Wikipedia documents. The first is Body First Selection (BFS) where one sentence from the first section of a Wikipedia page is randomly sampled and another passage from the same page is considered as its positive context. The other is Wiki Link Prediction (WLP) where the sentence is sampled the same way as in BFS, but the passage is sampled from another hyperlinked Wikipedia page. These paragraph-level pre-training tasks are pre-trained with a bi-encoder architecture to support the embedding-based dense retrieval. Experiments on several QA datasets showed that the pre-trained model significantly outperform the widely used BM25 algorithm and the MLM pre-trained models when fine-tuning with a limited number of labeled data. However, BFS and WLP heavily rely on the special structures of web documents (e.g., multiple paragraph segmentation and hyperlinks), which may hinder their application on a general text corpus.

Another one borrows the idea of information bottleneck theory (Tishby and Zaslavsky, 2015) which says a good representation is a maximally compressed mapping of the input on the output. The autoencoder architecture, which performs the compress-then-reconstruct operation to the input, naturally conforms the information bottleneck principle. Specifically, the general autoencoder consists of an encoder and a decoder, where the encoder maps the input text to representations and the decoder is trained to reconstruct the input text from the representations. Lu *et al.* (2021) found that the decoder may take shortcuts by exploiting language patterns using its access to previous tokens. Thus, the vanilla autoencoder is not able to provide high-quality sequence representations. They proposed SEED which pre-trains autoencoder-based language model with a weak decoder to avoid the bypass effect. By restricting the model capacity and the attention flexibility of the decoder, the encoder can provide better text representations for dense retrieval. Experiments on three tasks, including web search, news recommendation, and openQA, demonstrate that SEED is able to boost the effectiveness and few-shot ability significantly.

Model Architectures Due to the quadratic time and memory complexity of self-attention mechanism in vanilla Transformer, the input length of Transformer-based PTMs is always limited to 512. However, documents in IR collections are often longer than 512, so vanilla Transformer-based PTMs are unsuitable to process long documents. Some studies have investigated designing new architectures to adapt to the IR scenario. For example, Longformer (Beltagy *et al.*, 2020) proposed to use a combination of a local self-attention and a global attention to sparse the attention matrix. Sekulic *et al.* (2020) applied Longformer-based pre-trained models to document ranking. Yang *et al.* (2020) proposed Siamese Multi-depth Transformer-based Hierarchical (SMITH) Encoder to handle long document matching tasks. SMITH learns document representations by hierarchically aggregating the sentence representations from bottom to top. SMITH is pre-trained with a novel masked sentence block prediction task in addition to MLM task. Experiments show SMITH outperforms BERT on two document matching tasks by increasing maximum input text length from 512 to 2048.

To learn better text sequence representations, Gao and Callan (2021a) proposed Condenser which modifies the Transformer architecture by adding a short circuit from the lower layer to the higher layers. Specifically, for a Transformer model with 12 layers like BERT, they added additional 2 layers on top of the model and also added a short circuit from the 6th layer to the 13th layer. For the short circuit, the token representations from the 6th layer are directly input to the 13th layer and there is no input from the previous layer, i.e., the 12th layer, except for the special [CLS] token. They claim that the [CLS] token in the 7-12th layer will focus more on the global meaning of the input text to provide enough information for the top layers to predict the original tokens. Their experiments showed Condenser improves over standard LM by large margins on various text retrieval and similarity tasks.

6.2 Pre-training Interaction Models for IR

The relevance estimation between a query and a document is to determine whether the information contained in the document satisfy the information need behind the query. Such information could be either a small piece of text span or a long passage, which makes the relevance pattern varies significantly. The representation-focused models are hard to capture such diverse matching patterns by relying on some simple interaction functions in the last layer. An alternative way is to employ PTMs to directly model complicated interaction patterns from low-level features. Since existing PTMs pay more attention to the representation learning rather than the interaction learning in original pre-training objectives, researchers proposed different learning strategies to capture the query-document interactions by further pre-training PTMs in domain data. According to the objective used in different pre-training models, we divide them into two categories: 1) Weak Supervised Learning; 2) Self-supervised Learning.

6.2.1 Weak Supervised Learning

Weak supervised learning aim to learn machine learning models on noisy data. To be more specific, labels are automatically generated by other models not human beings. And the learning objective of weak supervision is often the same as the objective of the downstream task, that is, the learning objective of weak supervision in IR is the ranking objective. Once the models are pre-trained on the generated noisy data, they can also be fine-tuned with supervised training data on the target IR tasks (Dehghani *et al.*, 2017b; Luo *et al.*, 2017b; Zamani and Croft, 2018; Zamani *et al.*, 2018a; Zhang *et al.*, 2020b).

After the rise of NeuIR, researchers explored to pre-train a simple neural interaction model on weakly supervised data for ad-hoc retrieval to verify its effectiveness. Dehghani *et al.* (2017b) first investigated the weak supervised learning for IR. They train neural interaction models on billions of noisy training data automatically generated by BM25. The input is query-document pairs and the model architecture is a simple feed-forward neural network. Both pointwise learning and pairwise learning are studied under weak supervised setting. Experiments showed the trained neural model using weak supervision can outperform BM25. To study the reason, Zamani and Croft (2018) theoretically analyzed weak supervision from the perspective of the risk minimization framework to verify its effectiveness. Recently, Zhang *et al.* (2020b) proposed a reinforcement weak supervision method with BERT, called ReInfoSelect. ReInfoSelect trains a selector model to select some constructed anchor-document pairs for training the BERT-based ranker via reinforcement learning. It takes the ranking performance (i.e., NDCG) as the reward. Experiments showed the neural ranker trained by ReInfoSelect can match the effectiveness of neural rankers trained on private commercial search logs.

6.2.2 Self-supervised Learning

Self-supervised learning is somehow a blend of supervised learning and unsupervised learning (Liu *et al.*, 2021e; Qiu *et al.*, 2020). The basic idea of self-supervised learning is to predict any part of the input from other parts in some form, whose learning objective is not the same as the objective in the downstream tasks. So the labels of training data are often from the data itself rather than the same as in a specific task, like relevance judgments in IR. The learning paradigm of self-supervised learning is entirely the same as supervised learning. Recent PTMs on pre-training interaction models such as BERT and StructBERT, aim to learn the **coherence** relationship between two sentences by predicting the sentence order. Specifically, they usually take two sentences as input and pre-train the interaction model with Next Sentence Prediction (NSP) task or Sentence Order Prediction (SOP) task. However, the coherence relationship quite diverges from relevance, which is the most important requirement of IR. So, researchers designing PTMs tailored for IR mainly from the following two aspects: 1) Pre-training Objectives; 2) Model Architectures.

Pre-training Objectives Relevance is a vague notion in IR, so is there any other object to be a good proxy of relevance? Inspired by the query likelihood model (QL) (Ponte and Croft, 2017), Ma *et al.* (2021b) proposed a novel pre-training task named Representative wOrds Prediction (ROP) for ad-hoc retrieval, and the pre-trained model is called PROP. QL assumes that the query is a piece of representative text generated from the "ideal" document (Liu and Croft, 2006). Thus, modeling **representativeness** may benefit to capture the relevance between the query and the document. To verify this hypothesis, ROP samples pairs of word sets according to the multinomial unigram language model (Zhai, 2007), and then pre-trains the Transformer to predict the pairwise preference. Experiments show PROP outperforms other pre-trained models like BERT and ICT on a variety of ad-hoc retrieval tasks. Moreover, under both the zero-shot and few-shot settings, PROP can achieve surprising performance, and even outperform BM25 on Gov2 without fine-tuning. Ma *et al.* (2021c) further proposed B-PROP by leveraging BERT to replace the classical unigram language model for the ROP task construction. Inspired by the divergence-from-randomness idea (Amati and Rijsbergen, 2002), they proposed a contrastive method to leverage BERT's [CLS]-token attention to sample representative words. Experiments show B-PROP performs better than PROP on the downstream document ranking datasets. Ma *et al.* (2021d) proposed HARP with anchor texts and hyperlinks to replace the sampling method, as sampling may introduce noise to the data. Experimental results show that HARP can perform better than PROP on MS-MARCO Document Ranking and TREC DL. As most existing work adopts the two-stage training paradigm, models' off-the-shelf parameters can be largely updated in the fine-tuning process. What knowledge on earth do these models have learned still remains under-investigated. To this end, Chen *et al.* (2022) aimed to incorporate IR axioms into model pre-training and proposed a novel model named ARES. They generated training samples with specific IR axioms or heuristics to guide the training of ARES. Experimental results have shown the effectiveness of ARES, especially in low-resource scenarios where supervision data is limited.

Model Architectures Those works in Section 6.1.2 on designing PTMs for handling long texts can also be applied in pre-training interaction models. There is less effort on designing new interaction model architectures for IR as the self-attention mechanism of the Transformer architecture does provide a solution to do interaction between texts. 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, they can pre-compute the document representations and accelerate the inference for re-ranking.

6.3 Summary

Fine-tuning the Transformer-based PTMs has dominated almost every component in IR due to its convenience and effectiveness in recent years. However, the performance improvement on different IR tasks was still limited since original pre-training objectives are designed to learn the language coherence, e.g., predicting the masked token or the sentence order (Devlin *et al.*, 2019). To better leverage the pre-training paradigm for IR, there are two main lines of researches which concentrate effort on designing novel PTMs tailored for IR. The first one looks for novel pre-training objectives that better resemble IR requirements, e.g., the Inverse Cloze Task (Lee *et al.*, 2019b), the Wiki Link Prediction Chang *et al.*, 2020, and the representativeness of words prediction (Ma *et al.*, 2021b; Ma *et al.*, 2021c). Though different learning objectives are introduced and claimed to be beneficial to IR tasks. However, it still remains unclear how good these learning objectives satisfy the IR requirements for lacking of theoretical basis. Moreover, some of the pre-training objectives is strongly related to the weak learning since both of them rely on heuristic rules of IR, and the difference between this two learning strategies has been less studied. The second one focuses on designing new model architectures which aim to satisfy the heterogeneity structures in and between queries and documents, e.g., Longformer (Beltagy *et al.*, 2020) and SEED (Lu *et al.*, 2021). There are still very few works in this direction, and most of them have only made minor changes to original BERT model. This is due to the fact that the BERT model has been well trained on a very large-scale corpus, and a completely redesigned architecture leads to high model training cost. Moreover, it also requires in-depth analysis on the basis of the transformer architecture, and rethink the design criteria of architectures from the view of IR. Finally, the fundamental question to the design of both pre-training objectives and architectures lies at the concept of the relevance in IR. Based on this view, it highlights the need for more systematic research concerning the definition of the relevance instead of heuristic hands-on learning objectives or model architectures.

7

Resources of Pre-training Methods in IR

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

7.1 Datasets for Pre-Training

As discussed in Section 6, pre-training objectives designed for IR are mostly based on a (or more) large-scale collection(s). We thus consider the collections for pre-training tasks in IR with the following properties:

- **Large collection size**: In a broad sense, collection size is a 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, etc. These structures can be exploited in IR pre-training tasks to capture inter-page semantic relation. Moreover, hyperlinks between the pages(e.g., anchor-page linking and page-page linking) provide intra-page semantic relations, which can also be used in IR pre-training.

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

Dataset	Source	#Docs	Language	Latest crawl date
Books[1]	Book	74M	ENG	2015
C4[2]	web extracted text	0.3B	ENG	2019
Wikipedia[3]	Wiki text	10M	Multi-lang	monthly update
RealNews[4]	News	120GB	ENG	2019
Amazon[5]	reviews	11GB	ENG	2003
WT10G[6]	web pages	1.7M	ENG	1997
GOV2[7]	pages in .GOV	25M	ENG	2004
CWP200T	Chinese web pages	7B	CHN	2015
SogouT[8]	Sogou web pages	1.17B	CHN	2016
ClueWeb09[9]	web pages	1.04B	Multi-lang	2009
ClueWeb12[10]	web pages	0.73B	ENG	2012
MS MARCO[11]	Bing web pages	3.2M	ENG	2018

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

[1] https://github.com/huggingface/datasets/tree/master/datasets/bookcorpus

[2] https://github.com/huggingface/datasets/tree/master/datasets/c4

[3] https://dumps.wikimedia.org/

[4] https://github.com/rowanz/grover/tree/master/realnews

[5] https://snap.stanford.edu/data/web-Amazon.html

[6] http://ir.dcs.gla.ac.uk/test_collections/wt10g.html

[7] http://ir.dcs.gla.ac.uk/test_collections/access_to_data.html

[8] http://www.sogou.com/labs/resource/t.php

[9] https://lemurproject.org/clueweb09/

[10] https://lemurproject.org/clueweb12/

[11] https://microsoft.github.io/msmarco/

properties, the collection might be better for IR pre-training tasks. Given the suggested properties of a IR pre-training dataset, we sort out some public available datasets which are potentially useful 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*: This dataset aims to align books with the corresponding movie releases by associating the visual information with descriptive text. The text conveys both visual content (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).
    - *C4*: Colossal Clean Crawled Corpus (C4) is a dataset consisting of more than 300 GBs 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 in XML structure. It takes advantages in well-organized document structures, entity links, and rich information, which are suitable for pre-training tasks in IR.
    - *RealNews*: RealNews is a large-scale corpus containing news articles from Common Crawl. The documents are scraped from Common Crawl, limited to more than 5000 news domains indexed by Google News. News from Common Crawl dumps from December 2016 to March 2019 were used as training data; articles published in April 2019 were used for evaluation.

  - *Amazon Reviews*: This dataset consists of Amazon shopping reviews from amazon. The data spans a period of 18 years, including more than 35 million reviews up to March 2013. Reviews include user and product information, ratings, and a plaintext review.

- **IR related corpus**: These kind of corpus contain documents which are similar to downstream IR tasks. Pre-training on these corpus can further minimize the gap between pre-training and downstream IR tasks, providing a better opportunity to achieve better ranking performance.

  - *WT10G*: WT10G (Web Track 10Gigabytes) was collected 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 graph structure of web links, server size distribution, inclusion of inter-domain links and 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, text and the extracted text of pdf, word and postscript. The collection is about 426GB and contains 25 million documents. The large proportion of web pages has potential for pre-training tasks with text-based self-supervised learning objectives.
  - *CWP200T, SogouT*: CWP200T and SogouT (Luo *et al.*, 2017a) 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 10 languages which 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 consisting of 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[12] to further explore the available datasets for pre-training tasks. And, the corpus with web pages mostly contain two important relations (i.e., inter-document (e.g., html structure) and intra-document (e.g., hyperlinks, anchor-page links) relations). These relations provides implications to design different pre-training objectives for IR 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 and query-oriented tasks. The abbreviations of these tasks are further used in Table 7.2 as the potential tasks of different datasets. We introduce each specific task 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.

[12] https://github.com/huggingface/datasets/tree/master/datasets

  - *Personalized Search* (**PS**): User-specific Ranking.

- **Query-oriented**

  - *Query reformulation* (**QR**): Iiteratively modifying a query to improve the quality of search engine results in order to enhance user's search satisfaction.
  - *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 (or 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 extract the most used and most important terms in a document.

The detailed description of each collection is as follows:

1. Robust track (Voorhees, 2004) 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-scale collection of queries and documents. The final released dataset contains a four-level relevance judgement for each query-document pair.

Dataset	Subdata	Size	Source	Potential Tasks
Robust	Robust04	0.5M docs, 250 queries	TREC Robust Track	FSR, AR, QR
	Robust05	1M docs, 50 queries		
TREC MQ	MQ2007	6.5K docs, 1.7K queries	TREC Million Query track	FSR, AR, QR
	MQ2008	1.4K docs, 784 queries		
Clueweb	09-CatB	50M docs, 150 queries	Web pages	FSR, AR, QR, KE
	12-CatB	50M docs		
TREC web track	99-2014	See [13]	TREC web track	FSR, AR, QR
TREC DL track	2019-2021	See [14]	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)	\	1.8M docs	News articles	DS
Debatepedia	\	1,303 debates	Debate key points	SG, DS
DUC	2001-07	300 clusters, See [15]	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.

[13] https://trec.nist.gov/data/webmain.html

[14] https://microsoft.github.io/msmarco/TREC-Deep-Learning.html

[15] https://duc.nist.gov/data.html

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

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

11. BEIR (Benchmarking IR) (Thakur *et al.*, 2021) is a new heterogeneous benchmark containing different IR tasks. The benchmark contains 18 datasets covering 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 several neural retrieval methods.

12. 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.

13. 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 selected 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.

14. The CNN/Daily Mail dataset (See *et al.*, 2017) is a large-scale collection of news articles and further modified for summarization. It consists of more than 280,000 training samples and 11,490 test set samples. The documents in the training set have 29.74 sentences with 766 words on average while the summaries consist of 53 words and 3.72 sentences on average.

15. New York Times (NYT)[16] is a large-scale document summarization dataset. It contains well curated articles from *The New York Times* between 1987 and 2007. The summaries were written by library scientists, making it particularly useful as an extractive summarization dataset.

16. Debatepedia is collected from *debatepedia.org*. It is an encyclopedia of pro and con arguments and quotes on critical debate topics. There are totally 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 and documents per query is 5 and 4, respectively.

17. 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 of the article is 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 contains on average 10 documents. For the details of other versions, please refer to here[17].

18. 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.

[16] https://catalog.ldc.upenn.edu/LDC2008T19

[17] https://duc.nist.gov/data.html

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
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

Challenges and Future Work

In this chapter, we discuss current challenges and suggest some promising directions for pre-training methods researching in the IR field.

8.1 New Objectives & Architectures Tailored for IR

Although the general-purpose pre-trained language models are suitable for learning the universal language knowledge, designing the pre-training and tuning methods that more closely resemble downstream tasks is admittedly a more efficient way to obtain better performance on specific tasks (Zhang *et al.*, 2020a; Ke *et al.*, 2019). 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 exploration 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.*, 2020; Guu *et al.*, 2020; Ma *et al.*, 2021b; Ma *et al.*, 2021c; Liu *et al.*, 2021g; Ma *et al.*, 2021d) on the pre-training objectives tailored for IR. For example, Lee *et al.* (2019b) proposed to pre-train with a large-scale document collection with the Inverse Cloze Task (ICT) for retrieval tasks. Besides ICT, Chang *et al.* (2020) also proposed to capture the inner-page and inter-page semantic relations with Body First Selection (BFS) and Wiki Link Prediction (WLP) for passage retrieval in QA tasks. For the re-ranking component, Ma *et al.* (2021b) and Ma *et al.* (2021c) proposed the Representative Words Prediction (ROP) objective for pre-training, which achieves significant improvement. In addition to constructing pseudo query-document pairs from the raw text, 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 for pre-training. 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 ad-hoc retrieval datasets. On the whole, the underlying idea of all these pre-training objectives tailored for IR is to simulate the relevance relationship between queries and documents. However, it is still in the preliminary stage to design more 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 dual-encoder architecture 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 tasks. 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 stable improvement over standard LM and shows comparable performance to strong task-specific PTMs. Similarly, in order to obtain better document embeddings for dense retrieval, Lu *et al.* (2021) presented a new auto-encoder architecture with restricted attention flexibility. Based on this, the new architecture could create an information bottleneck in the auto-encoder and force the encoder to provide better document representations. However, compared with attempts to investigate new pre-training objectives for IR, designing an ingenious pre-training model architecture which is suitable for IR tasks has not been well explored.

Beyond Fine-Tuning. Up to now, fine-tuning is the most dominant method to apply 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, the emergence of GPT-3 (Brown *et al.*, 2020) makes the prompt tuning (Liu *et al.*, 2021c) attract more research attention. Prompt tuning needs to design discrete (Petroni *et al.*, 2019; Gao *et al.*, 2021c) or continuous (Liu *et al.*, 2021f; Lester *et al.*, 2021) prompts for specific downstream tasks. For now, it is a promising way to reduce the computational cost of using pre-trained models for downstream tasks. In fact, prompt tuning 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, 2021a), and fact probing (Petroni *et al.*, 2019; Jiang *et al.*, 2020). However, there has been no mature work on prompt tuning for IR tasks. From another perspective, the design of most of existing PTMs is driven by the fine-tuning paradigm, but it is unclear whether the exploring of different PTMs will produce pre-trained models which are more effective when they are used with prompt tuning to solve IR tasks.

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

Developing PTMs based on 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 model pre-training, and on the other hand, incorporating extra data has great potential to enhance document representations for IR tasks.

Multi-modal Pre-Training for IR. Large-scale pre-training methods have been widely developed with diverse real-world modalities (e.g., text, image, audio, and video) and different practical applications. In recent years, there has been an upsurging interest in cross-modal tasks, e.g., image-text retrieval (Lee *et al.*, 2018; 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, PTMs based on cross modalities also have improved research interests, such as image-text (Lu *et al.*, 2019; Li *et al.*, 2020b), video-text (Sun *et al.*, 2019a), or audio-text (Chuang *et al.*, 2020). Among the Vision-and-Language pre-training (VLP) research, most current works focus on the interaction of images and texts (Li *et al.*, 2020b; Su *et al.*, 2020b; Lu *et al.*, 2019; Li *et al.*, 2020d), 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 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 UNIMO (Li *et al.*, 2021) models textual knowledge and visual knowledge in a unified semantic space and results in improved performance for NLP tasks. However, most of these works are not evaluated on IR tasks. Besides, although multi-modal PTMs has made great progress in recent years, Cao *et al.* (2020) proved that the textual modality is more dominant than image during the multi-modal pre-training process. Based on this, the benefits of cross-modal learning are mainly reflected on image-based tasks. Thus, it is worth further exploring to design better vision-language pre-training objectives pointing at IR tasks. 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 easily available. However, due to the cost and required dataset, pre-training large language models for each language is not practical. Specially, the large-scale annotations are hard to obtain for low-resource languages. Additionally, some empirical results show that training one model with several languages could get better performance on some tasks than training several monolingual models independently (Conneau and Lample, 2019; Ni *et al.*, 2021b). Hence, training a language model based on multi-lingual data may be a good attempt for IR tasks. In fact, some existing multi-lingual pre-trained models, such as mBERT (Devlin *et al.*, 2019), XLM (Conneau and Lample, 2019), and Unicoder (Huang *et al.*, 2019), have shown their language transfer abilities over a wide range of tasks (Wu and Dredze, 2019). For example, Shi *et al.* (2020) constructed the re-ranking model for non-English corpus based on the mBERT, aiming to leverage the relevance information learned in English. They found that this significantly improves search quality for non-English 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 provide a good prior for model training. Thus, introducing external knowledge into PTMs to get knowledge-enhanced representations for IR is another research line. Based on knowledge graphs, there have been many explorations to integrate entity and relation embeddings or their alignments into pre-trained models training (Zhang *et al.*, 2019c; Sun *et al.*, 2019b; Wang *et al.*, 2021). Different from structured knowledge, unstructured knowledge, e.g., the domain-specific data, is more abundant but also noisier. Several works (Beltagy *et al.*, 2019; Lee *et al.*, 2019a) have attempted to further training the general pre-trained models on these data to get better performance for specific domains or tasks. However, most of these efforts are not tailored for IR. In the future, how to effectively model these knowledge for IR needs to be further explored. On the other hand, all existing works store knowledge with model parameters implicitly. How to model 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 paradigm has some disadvantages in practical scenarios, which will produce sub-optimal performance. Recently, the application of PTMs in the retrieval component 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 model. However, advances in PTMs-based retrieval models resulting in a shift from the inverted index towards the dense vector-based index makes the joint training possible. In fact, there have been studies (Zhang *et al.*, 2021a; Zhan *et al.*, 2021a; Zhan *et al.*, 2022) to explore the joint training of retrieval models and the index module. In this way, the index building can benefit from the relevance information between queries and documents directly. In addition to the profits from the joint learning of index and retrieval, there have been works finding that it is beneficial to train retrievers and re-rankers in a correlated manner. For example, the retriever can be improved by distilling knowledge from the re-ranker (Qu *et al.*, 2021; Hofstätter *et al.*, 2020), and the re-ranker can be improved with hard negatives 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 optimize two components jointly and contribute to the final ranking performance. Nevertheless, these works are only preliminary attempts in this direction. In fact, the joint learning of two components, i.e., retrieval and re-ranking, cannot be implemented trivially and many problems have not been solved well. Besides, researchers in this field have not ventured into the end-to-end learning of the whole pipeline, including indexing, retrieval, and re-ranking.

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

Beyond the traditional multi-stage IR systems, the state-of-the-art pre-trained models 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, it is possible to set about the next generation of IR systems.

Metzler *et al.* (2021) proposed a vision to build model-based IR system based on the powerful pre-trained models. Within the framework, the index is embedded into the model itself during the model training process, and retrieval and re-ranking components are implemented integrately with model inference. However, this work only gives a beautiful vision and vague framework. Recently, Tay *et al.* (2022) implemented this new IR paradigm based on the T5 model. The significant performance is achieved by training the model with indexing (i.e., documents to docids) and retrieval (i.e., queries to docids) in a multi-task setup. At about the same time, Zhou *et al.* (2022) presented DynamicRetriever, which builds the model-based IR system based on BERT. They firstly fine-tuned the BERT-based dense retriever with query-document pairs, and then initialized the model parameters, especially the projection matrix with generated document embeddings. Finally, the model is further fine-tuned with query-docid pairs. Nevertheless, these works are only preliminary explorations and there are still many deficiencies to be improved. For example, how to build the semantics-based document identifications, and how to update the model when the document collection changes? Besides, there are a number of challenges needing to be solved before the model-based IR system can be applied in practice. At present, the capacity of existing pre-trained models is limited. For example, they do not have a real understanding of world knowledge, and it is challenging for them to develop the reasoning ability (e.g., arithmetic, logic, etc). Moreover, it is desiderative that the model-based IR system could be interpretable, debuggable and controllable. In fact, this is a core issue that all neural-based models need to address before they are applied.

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

This work was funded by the National Natural Science Foundation of China (NSFC) under Grants No. 61902381, No. 62006218, and No. 61732008, the Youth Innovation Promotion Association CAS under Grants No. 2021100, and 20144310, the Young Elite Scientist Sponsorship Program by CAST under Grants No. YESS20200121, the Lenovo-CAS Joint Lab Youth Scientist Project, and the Tsinghua University Guoqiang Research Institute.

References

Agirre, E., X. Arregi, and A. Otegi. (2010). “Document Expansion Based on WordNet for Robust IR”. In: *COLING 2010, 23rd International Conference on Computational Linguistics, Posters Volume, 23-27 August 2010, Beijing, China*. Ed. by C. Huang and D. Jurafsky. Chinese Information Processing Society of China. 9–17. URL: https://aclanthology.org/C10-2002/.

Agosti, M., S. Marchesin, and G. Silvello. (2020). “Learning Unsupervised Knowledge-Enhanced Representations to Reduce the Semantic Gap in Information Retrieval”. *ACM Transactions on Information Systems*. 38(4): 1–48. DOI: 10.1145/3417996. URL: https://doi.org/10.1145/3417996.

Ai, Q., L. Yang, J. Guo, and W. B. Croft. (2016a). “Analysis of the Paragraph Vector Model for Information Retrieval”. In: *Proceedings of the 2016 ACM International Conference on the Theory of Information Retrieval*. ACM. DOI: 10.1145/2970398.2970409. URL: https://doi.org/10.1145/2970398.2970409.

Ai, Q., L. Yang, J. Guo, and W. B. Croft. (2016b). “Improving Language Estimation with the Paragraph Vector Model for Ad-hoc Retrieval”. In: *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*. ACM. DOI: 10.1145/2911451.2914688. URL: https://doi.org/10.1145/2911451.2914688.

Alberti, C., J. Ling, M. Collins, and D. Reitter. (2019). "Fusion of Detected Objects in Text for Visual Question Answering". In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Association for Computational Linguistics. DOI: 10.18653/v1/d19-1219. URL: https://doi.org/10.18653/v1/d19-1219.

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*. ACM. DOI: 10.1145/3331184.3331265. URL: https://doi.org/10.1145/3331184.3331265.

Amati, G. and C. J. V. Rijsbergen. (2002). "Probabilistic models of information retrieval based on measuring the divergence from randomness". *ACM Transactions on Information Systems*. 20(4): 357–389. DOI: 10.1145/582415.582416. URL: https://doi.org/10.1145/582415.582416.

Amer, N. O., P. Mulhem, and M. Géry. (2016). "Toward Word Embedding for Personalized Information Retrieval". *CoRR*. abs/1606.06991. arXiv: 1606.06991. URL: http://arxiv.org/abs/1606.06991.

Antol, S., A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. L. Zitnick, and D. Parikh. (2015). "VQA: Visual Question Answering". In: *2015 IEEE International Conference on Computer Vision (ICCV)*. IEEE. DOI: 10.1109/iccv.2015.279. URL: https://doi.org/10.1109/iccv.2015.279.

Aumüller, M., E. Bernhardsson, and A. Faithfull. (2020). "ANN-Benchmarks: A benchmarking tool for approximate nearest neighbor algorithms". *Information Systems*. 87(Jan.): 101374. DOI: 10.1016/j.is.2019.02.006. URL: https://doi.org/10.1016/j.is.2019.02.006.

Bae, S., T. Kim, J. Kim, and S. Lee. (2019). "Summary Level Training of Sentence Rewriting for Abstractive Summarization". *CoRR*. abs/1909.08752. arXiv: 1909.08752. URL: http://arxiv.org/abs/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”. *CoRR*. abs/2010.00768. arXiv: 2010.00768. URL: https://arxiv.org/abs/2010.00768.

Bao, H., L. Dong, F. Wei, W. Wang, N. Yang, X. Liu, Y. Wang, J. Gao, S. Piao, M. Zhou, and H. Hon. (2020). “UniLMv2: Pseudo-Masked Language Models for Unified Language Model Pre-Training”. In: *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*. Vol. 119. *Proceedings of Machine Learning Research*. PMLR. 642–652. URL: http://proceedings.mlr.press/v119/bao20a.html.

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”. *CoRR*. abs/1801.07704. arXiv: 1801.07704. URL: http://arxiv.org/abs/1801.07704.

Beltagy, I., K. Lo, and A. Cohan. (2019). “SciBERT: A Pretrained Language Model for Scientific Text”. In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Association for Computational Linguistics. DOI: 10.18653/v1/d19-1371. URL: https://doi.org/10.18653/v1/d19-1371.

Beltagy, I., M. E. Peters, and A. Cohan. (2020). “Longformer: The Long-Document Transformer”. *CoRR*. abs/2004.05150. arXiv: 2004.05150. URL: https://arxiv.org/abs/2004.05150.

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. DOI: 10.1109/tpami.2013.50. URL: https://doi.org/10.1109/tpami.2013.50.

Bi, K., Q. Ai, and W. B. Croft. (2020). “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*. ACM. DOI: 10.1145/3397271.3401192. URL: https://doi.org/10.1145/3397271.3401192.

Bi, K., Q. Ai, and W. B. Croft. (2021a). “Asking Clarifying Questions Based on Negative Feedback in Conversational Search”. In: *Proceedings of the 2021 ACM SIGIR International Conference on Theory of Information Retrieval.* ACM. DOI: 10.1145/3471158.3472232. URL: https://doi.org/10.1145/3471158.3472232.

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.* ACM. DOI: 10.1145/3404835.3462911. URL: https://doi.org/10.1145/3404835.3462911.

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.* ACM. DOI: 10.1145/3442381.3450110. URL: https://doi.org/10.1145/3442381.3450110.

Bojanowski, P., E. Grave, A. Joulin, and T. Mikolov. (2017). “Enriching Word Vectors with Subword Information”. *Trans. Assoc. Comput. Linguistics.* 5: 135–146. URL: https://transacl.org/ojs/index.php/tacl/article/view/999.

Bromley, J., I. Guyon, Y. LeCun, E. Säckinger, and R. Shah. (1993). “Signature Verification Using a Siamese Time Delay Neural Network”. In: *Advances in Neural Information Processing Systems 6, [7th NIPS Conference, Denver, Colorado, USA, 1993].* Ed. by J. D. Cowan, G. Tesauro, and J. Alspector. Morgan Kaufmann. 737–744. URL: http://papers.nips.cc/paper/769-signature-verification-using-a-siamese-time-delay-neural-network.

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. (2020). "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. Balcan, and H. 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. (2005). "Learning to rank using gradient descent". In: *Proceedings of the 22nd international conference on Machine learning - ICML '05*. ACM Press. DOI: 10.1145/1102351.1102363. URL: https://doi.org/10.1145/1102351.1102363.

Burges, C. J. C., R. Ragno, and Q. V. Le. (2006). "Learning to Rank with Nonsmooth Cost Functions". In: *Advances in Neural Information Processing Systems 19, Proceedings of the Twentieth Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 4-7, 2006*. Ed. by B. Schölkopf, J. C. Platt, and T. Hofmann. MIT Press. 193–200. URL: https://proceedings.neurips.cc/paper/2006/hash/af44c4c56f385c43f2529f9b1b018f6a-Abstract.html.

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: *Computer Vision – ECCV 2020*. Springer International Publishing. 565–580. DOI: 10.1007/978-3-030-58539-6_34. URL: https://doi.org/10.1007/978-3-030-58539-6_34.

Carbinell, J. and J. Goldstein. (2017). "The Use of MMR, Diversity-Based Reranking for Reordering Documents and Producing Summaries". *ACM SIGIR Forum*. 51(2): 209–210. DOI: 10.1145/3130348.3130369. URL: https://doi.org/10.1145/3130348.3130369.

Chang, W., F. X. Yu, Y. Chang, Y. Yang, and S. Kumar. (2020). "Pre-training Tasks for Embedding-based Large-scale Retrieval". In: *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net. URL: https://openreview.net/forum?id=rkg-mA4FDr.

Chen, J., Y. Liu, Y. Fang, J. Mao, H. Fang, S. Yang, X. Xie, M. Zhang, and S. Ma. (2022). "Axiomatically Regularized Pre-training for Ad hoc Search". In: *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*.

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*. 39(3): 1–35. DOI: 10.1145/3448127. URL: https://doi.org/10.1145/3448127.

Chen, R.-C. and C.-J. Lee. (2020). "Incorporating Behavioral Hypotheses for Query Generation". In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics. DOI: 10.18653/v1/2020.emnlp-main.251. URL: https://doi.org/10.18653/v1/2020.emnlp-main.251.

Chen, W.-F., S. Syed, B. Stein, M. Hagen, and M. Potthast. (2020a). "Abstractive Snippet Generation". In: *Proceedings of The Web Conference 2020*. ACM. DOI: 10.1145/3366423.3380206. URL: https://doi.org/10.1145/3366423.3380206.

Chen, X., N. Zhang, X. Xie, S. Deng, Y. Yao, C. Tan, F. Huang, L. Si, and H. Chen. (2021b). "KnowPrompt: Knowledge-aware Prompt-tuning with Synergistic Optimization for Relation Extraction". *CoRR*. abs/2104.07650. arXiv: 2104.07650. URL: https://arxiv.org/abs/2104.07650.

Chen, X., B. He, K. Hui, L. Sun, and Y. Sun. (2021c). "Simplified TinyBERT: Knowledge Distillation for Document Retrieval". In: *Lecture Notes in Computer Science*. Springer International Publishing. 241–248. DOI: 10.1007/978-3-030-72240-1_21. URL: https://doi.org/10.1007/978-3-030-72240-1_21.

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. DOI: 10.1109/icassp40776.2020.9053531. URL: https://doi.org/10.1109/icassp40776.2020.9053531.

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 - WWW '05*. ACM Press. DOI: 10.1145/1062745.1062807. URL: https://doi.org/10.1145/1062745.1062807.

Choi, J., E. Jung, J. Suh, and W. Rhee. (2021). "Improving Bi-encoder Document Ranking Models with Two Rankers and Multi-teacher Distillation". In: *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM. DOI: 10.1145/3404835.3463076. URL: https://doi.org/10.1145/3404835.3463076.

Chuang, Y.-S., C.-L. Liu, H.-y. Lee, and L.-s. Lee. (2020). "SpeechBERT: An Audio-and-Text Jointly Learned Language Model for End-to-End Spoken Question Answering". In: *Interspeech 2020*. ISCA. DOI: 10.21437/interspeech.2020-1570. URL: https://doi.org/10.21437/interspeech.2020-1570.

Clinchant, S. and F. Perronnin. (2013). "Aggregating Continuous Word Embeddings for Information Retrieval". In: *Proceedings of the Workshop on Continuous Vector Space Models and their Compositionality, CVSM@ACL 2013, Sofia, Bulgaria, August 9, 2013*. Ed. by A. Allauzen, H. Larochelle, C. D. Manning, and R. Socher. Association for Computational Linguistics. 100–109. URL: https://aclanthology.org/W13-3212/.

Conneau, A. and D. Kiela. (2018). “SentEval: An Evaluation Toolkit for Universal Sentence Representations”. In: *Proceedings of the Eleventh International Conference on Language Resources and Evaluation, LREC 2018, Miyazaki, Japan, May 7-12, 2018*. Ed. by N. Calzolari, K. Choukri, C. Cieri, T. Declerck, S. Goggi, K. Hasida, H. Isahara, B. Maegaard, J. Mariani, H. Mazo, A. Moreno, J. Odijk, S. Piperidis, and T. Tokunaga. European Language Resources Association (ELRA). URL: http://www.lrec-conf.org/proceedings/lrec2018/summaries/757.html.

Conneau, A. and G. Lample. (2019). “Cross-lingual Language Model Pretraining”. In: *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*. Ed. by H. M. Wallach, H. Larochelle, A. Beygelzimer, F. d’Alché-Buc, E. B. Fox, and R. Garnett. 7057–7067. URL: https://proceedings.neurips.cc/paper/2019/hash/c04c19c2c2474dbf5f7ac4372c5b9af1-Abstract.html.

Crammer, K. and Y. Singer. (2001). “Pranking with Ranking”. In: *Advances in Neural Information Processing Systems 14 [Neural Information Processing Systems: Natural and Synthetic, NIPS 2001, December 3-8, 2001, Vancouver, British Columbia, Canada]*. Ed. by T. G. Dietterich, S. Becker, and Z. Ghahramani. MIT Press. 641–647. URL: https://proceedings.neurips.cc/paper/2001/hash/5531a5834816222280f20d1ef9e95f69-Abstract.html.

Craswell, N., W. B. Croft, J. Guo, B. Mitra, and M. de Rijke. (2017). “Report on the SIGIR 2016 Workshop on Neural Information Retrieval (Neu-IR)”. *ACM SIGIR Forum*. 50(2): 96–103. DOI: 10.1145/3053408.3053425. URL: https://doi.org/10.1145/3053408.3053425.

Craswell, N., B. Mitra, E. Yilmaz, D. Campos, and J. Lin. (2021). “MS MARCO: Benchmarking Ranking Models in the Large-Data Regime”. In: *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM. DOI: 10.1145/3404835.3462804. URL: https://doi.org/10.1145/3404835.3462804.

Croft, W. B. and J. D. Lafferty. (2003). “Language Modeling for Information Retrieval”. In: *The Springer International Series on Information Retrieval*.

Dai, Z. and J. Callan. (2019a). “Context-Aware Sentence/Passage Term Importance Estimation For First Stage Retrieval”. *CoRR*. abs/1910.10687. arXiv: 1910.10687. URL: http://arxiv.org/abs/1910.10687.

Dai, Z. and J. Callan. (2019b). “Deeper Text Understanding for IR with Contextual Neural Language Modeling”. In: *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM. DOI: 10.1145/3331184.3331303. URL: https://doi.org/10.1145/3331184.3331303.

Dai, Z. and J. Callan. (2020a). “Context-Aware Document Term Weighting for Ad-Hoc Search”. In: *Proceedings of The Web Conference 2020*. ACM. DOI: 10.1145/3366423.3380258. URL: https://doi.org/10.1145/3366423.3380258.

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*. ACM. DOI: 10.1145/3397271.3401204. URL: https://doi.org/10.1145/3397271.3401204.

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

Dehghani, M., S. Rothe, E. Alfonseca, and P. Fleury. (2017a). “Learning to Attend, Copy, and Generate for Session-Based Query Suggestion”. In: *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. ACM. DOI: 10.1145/3132847.3133010. URL: https://doi.org/10.1145/3132847.3133010.

Dehghani, M., H. Zamani, A. Severyn, J. Kamps, and W. B. Croft. (2017b). “Neural Ranking Models with Weak Supervision”. In: *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM. DOI: 10.1145/3077136.3080832. URL: https://doi.org/10.1145/3077136.3080832.

Devlin, J., M.-W. Chang, K. Lee, and K. Toutanova. (2019). "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". In: *Proceedings of the 2019 Conference of the North*. Association for Computational Linguistics. DOI: 10.18653/v1/n19-1423. URL: https://doi.org/10.18653/v1/n19-1423.

Diaz, F., B. Mitra, and N. Craswell. (2016). "Query Expansion with Locally-Trained Word Embeddings". In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics. DOI: 10.18653/v1/p16-1035. URL: https://doi.org/10.18653/v1/p16-1035.

Dietz, L., M. Verma, F. Radlinski, and N. Craswell. (2017). "TREC Complex Answer Retrieval Overview". In: *Proceedings of The Twenty-Sixth Text REtrieval Conference, TREC 2017, Gaithersburg, Maryland, USA, November 15-17, 2017*. Ed. by E. M. Voorhees and A. Ellis. Vol. 500-324. *NIST Special Publication*. National Institute of Standards and Technology (NIST). URL: https://trec.nist.gov/pubs/trec26/papers/Overview-CAR.pdf.

Dou, Z.-Y., P. Liu, H. Hayashi, Z. Jiang, and G. Neubig. (2021). "GSum: A General Framework for Guided Neural Abstractive Summarization". In: *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics. DOI: 10.18653/v1/2021.naacl-main.384. URL: https://doi.org/10.18653/v1/2021.naacl-main.384.

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.

Echihabi, K., K. Zoumpatianos, T. Palpanas, and H. Benbrahim. (2019). "Return of the Lernaean Hydra". *Proceedings of the VLDB Endowment*. 13(3): 403–420. DOI: 10.14778/3368289.3368303. URL: https://doi.org/10.14778/3368289.3368303.

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 - SIGIR ’12*. ACM Press. DOI: 10.1145/2348283.2348405. URL: https://doi.org/10.1145/2348283.2348405.

Erkan, G. and D. R. Radev. (2004). “LexRank: Graph-based Lexical Centrality as Salience in Text Summarization”. *J. Artif. Intell. Res.* 22: 457–479. DOI: 10.1613/jair.1523. URL: https://doi.org/10.1613/jair.1523.

Fan, Y., J. Guo, X. Ma, R. Zhang, Y. Lan, and X. Cheng. (2021). “A Linguistic Study on Relevance Modeling in Information Retrieval”. In: *Proceedings of the Web Conference 2021*. ACM. DOI: 10.1145/3442381.3450009. URL: https://doi.org/10.1145/3442381.3450009.

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*. Association for Computational Linguistics. 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*. ACM. DOI: 10.1145/3077136.3080690. URL: https://doi.org/10.1145/3077136.3080690.

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*. ACM. DOI: 10.1145/3404835.3463098. URL: https://doi.org/10.1145/3404835.3463098.

Frej, J., P. Mulhem, D. Schwab, and J.-P. Chevallet. (2020). “Learning Term Discrimination”. In: *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM. DOI: 10.1145/3397271.3401211. URL: https://doi.org/10.1145/3397271.3401211.

Frieder, O., D. A. Grossman, A. Chowdhury, and G. Frieder. (2000). “Efficiency Considerations for Scalable Information Retrieval Servers”. *J. Digit. Inf.* 1(5). URL: http://journals.tdl.org/jodi/article/view/21.

Ganesh, P., Y. Chen, X. Lou, M. A. Khan, Y. Yang, D. Chen, M. Winslett, H. Sajjad, and P. Nakov. (2020). “Compressing Large-Scale Transformer-Based Models: A Case Study on BERT”. *CoRR*. abs/2002.11985. arXiv: 2002.11985. URL: https://arxiv.org/abs/2002.11985.

Ganguly, D., D. Roy, M. Mitra, and G. J. Jones. (2015). “Word Embedding based Generalized Language Model for Information Retrieval”. In: *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM. DOI: 10.1145/2766462.2767780. URL: https://doi.org/10.1145/2766462.2767780.

Gao, L. and J. Callan. (2021a). “Condenser: a Pre-training Architecture for Dense Retrieval”. In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics. DOI: 10.18653/v1/2021.emnlp-main.75. URL: https://doi.org/10.18653/v1/2021.emnlp-main.75.

Gao, L. and J. Callan. (2021b). “Is Your Language Model Ready for Dense Representation Fine-tuning?” *CoRR*. abs/2104.08253. arXiv: 2104.08253. URL: https://arxiv.org/abs/2104.08253.

Gao, L., Z. Dai, and J. Callan. (2020a). "Modularized Transfomer-based Ranking Framework". In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics. DOI: 10.18653/v1/2020.emnlp-main.342. URL: https://doi.org/10.18653/v1/2020.emnlp-main.342.

Gao, L., Z. Dai, and J. Callan. (2020b). "Understanding BERT Rankers Under Distillation". In: *Proceedings of the 2020 ACM SIGIR on International Conference on Theory of Information Retrieval*. ACM. DOI: 10.1145/3409256.3409838. URL: https://doi.org/10.1145/3409256.3409838.

Gao, L., Z. Dai, and J. Callan. (2021a). "COIL: Revisit Exact Lexical Match in Information Retrieval with Contextualized Inverted List". In: *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics. DOI: 10.18653/v1/2021.naacl-main.241. URL: https://doi.org/10.18653/v1/2021.naacl-main.241.

Gao, L., Z. Dai, and J. Callan. (2021b). "Rethink Training of BERT Rerankers in Multi-stage Retrieval Pipeline". In: *Lecture Notes in Computer Science*. Springer International Publishing. 280–286. DOI: 10.1007/978-3-030-72240-1_26. URL: https://doi.org/10.1007/978-3-030-72240-1_26.

Gao, L., Z. Dai, Z. Fan, and J. Callan. (2020c). "Complementing Lexical Retrieval with Semantic Residual Embedding". *CoRR*. abs/2004.13969. arXiv: 2004.13969. URL: https://arxiv.org/abs/2004.13969.

Gao, T., A. Fisch, and D. Chen. (2021c). "Making Pre-trained Language Models Better Few-shot Learners". In: *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Association for Computational Linguistics. DOI: 10.18653/v1/2021.acl-long.295. URL: https://doi.org/10.18653/v1/2021.acl-long.295.

Gillick, D., A. Presta, and G. S. Tomar. (2018). "End-to-End Retrieval in Continuous Space". *CoRR*. abs/1811.08008. arXiv: 1811.08008. URL: http://arxiv.org/abs/1811.08008.

Grbovic, M., N. Djuric, V. Radosavljevic, F. Silvestri, R. Baeza-Yates, A. Feng, E. Ordentlich, L. Yang, and G. Owens. (2016). "Scalable Semantic Matching of Queries to Ads in Sponsored Search Advertising". In: *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*. ACM. DOI: 10.1145/2911451.2911538. URL: https://doi.org/10.1145/2911451.2911538.

Grbovic, M., N. Djuric, V. Radosavljevic, F. Silvestri, and N. Bhamidipati. (2015). "Context- and Content-aware Embeddings for Query Rewriting in Sponsored Search". In: *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM. DOI: 10.1145/2766462.2767709. URL: https://doi.org/10.1145/2766462.2767709.

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*. ACM. DOI: 10.1145/2939672.2939754. URL: https://doi.org/10.1145/2939672.2939754.

Guo, J., Y. Cai, Y. Fan, F. Sun, R. Zhang, and X. Cheng. (2022). "Semantic Models for the First-Stage Retrieval: A Comprehensive Review". *ACM Transactions on Information Systems*. 40(4): 1–42. DOI: 10.1145/3486250. URL: https://doi.org/10.1145/3486250.

Guo, J., Y. Fan, Q. Ai, and W. B. Croft. (2016). "A Deep Relevance Matching Model for Ad-hoc Retrieval". In: *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*. ACM. DOI: 10.1145/2983323.2983769. URL: https://doi.org/10.1145/2983323.2983769.

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

Guu, K., K. Lee, Z. Tung, P. Pasupat, and M. Chang. (2020). "REALM: Retrieval-Augmented Language Model Pre-Training". *CoRR*. abs/2002.08909. arXiv: 2002.08909. URL: https://arxiv.org/abs/2002.08909.

Gysel, C. V., M. de Rijke, and E. Kanoulas. (2018). “Neural Vector Spaces for Unsupervised Information Retrieval”. *ACM Transactions on Information Systems*. 36(4): 1–25. DOI: 10.1145/3196826. URL: https://doi.org/10.1145/3196826.

Habibi, M., P. Mahdabi, and A. Popescu-Belis. (2016). “Question answering in conversations: Query refinement using contextual and semantic information”. *Data and Knowledge Engineering*. 106(Nov.): 38–51. DOI: 10.1016/j.datak.2016.06.003. URL: https://doi.org/10.1016/j.datak.2016.06.003.

Han, X., W. Zhao, N. Ding, Z. Liu, and M. Sun. (2021a). “PTR: Prompt Tuning with Rules for Text Classification”. *CoRR*. abs/2105.11259. arXiv: 2105.11259. URL: https://arxiv.org/abs/2105.11259.

Han, Y., G. Huang, S. Song, L. Yang, H. Wang, and Y. Wang. (2021b). “Dynamic Neural Networks: A Survey”. *CoRR*. abs/2102.04906. arXiv: 2102.04906. URL: https://arxiv.org/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*. ACM. DOI: 10.1145/3397271.3401061. URL: https://doi.org/10.1145/3397271.3401061.

Herbrich, R. (1999). “Support vector learning for ordinal regression”. In: *9th International Conference on Artificial Neural Networks: ICANN ’99*. IEE. DOI: 10.1049/cp:19991091. URL: https://doi.org/10.1049/cp:19991091.

Hinton, G. E., O. Vinyals, and J. Dean. (2015). “Distilling the Knowledge in a Neural Network”. *CoRR*. abs/1503.02531. arXiv: 1503.02531. URL: http://arxiv.org/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”. *CoRR*. abs/2010.02666. arXiv: 2010.02666. URL: https://arxiv.org/abs/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". In: *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM. DOI: 10.1145/3404835.3462891. URL: https://doi.org/10.1145/3404835.3462891.

Hofstätter, S., B. Mitra, H. Zamani, N. Craswell, and A. Hanbury. (2021b). "Intra-Document Cascading: Learning to Select Passages for Neural Document Ranking". In: *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM. DOI: 10.1145/3404835.3462889. URL: https://doi.org/10.1145/3404835.3462889.

Houlsby, N., A. Giurgiu, S. Jastrzebski, B. Morrone, Q. de Laroussilhe, A. Gesmundo, M. Attariyan, and S. Gelly. (2019). "Parameter-Efficient Transfer Learning for NLP". In: *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*. Ed. by K. Chaudhuri and R. Salakhutdinov. Vol. 97. *Proceedings of Machine Learning Research*. PMLR. 2790–2799. URL: http://proceedings.mlr.press/v97/houlsby19a.html.

Howard, J. and S. Ruder. (2018). "Universal Language Model Fine-tuning for Text Classification". In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics. DOI: 10.18653/v1/p18-1031. URL: https://doi.org/10.18653/v1/p18-1031.

Hu, B., Z. Lu, H. Li, and Q. Chen. (2014). "Convolutional Neural Network Architectures for Matching Natural Language Sentences". In: *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*. Ed. by Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger. 2042–2050. URL: https://proceedings.neurips.cc/paper/2014/hash/b9d487a30398d42ecff55c228ed5652b-Abstract.html.

Hu, E. J., Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, and W. Chen. (2021). "LoRA: Low-Rank Adaptation of Large Language Models". *CoRR*. abs/2106.09685. arXiv: 2106.09685. URL: https://arxiv.org/abs/2106.09685.

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". In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Association for Computational Linguistics. DOI: 10.18653/v1/d19-1252. URL: https://doi.org/10.18653/v1/d19-1252.

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 and Data Mining*. ACM. DOI: 10.1145/3394486.3403305. URL: https://doi.org/10.1145/3394486.3403305.

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 Conference on information and knowledge management - CIKM '13*. ACM Press. DOI: 10.1145/2505515.2505665. URL: https://doi.org/10.1145/2505515.2505665.

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*. 460(Oct.): 166–180. DOI: 10.1016/j.neucom.2021.07.013. URL: https://doi.org/10.1016/j.neucom.2021.07.013.

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*.

Jaleel, N. A., J. Allan, W. B. Croft, F. Diaz, L. S. Larkey, X. Li, M. D. Smucker, and C. Wade. (2004). "UMass at TREC 2004: Novelty and HARD". In: *Proceedings of the Thirteenth Text REtrieval Conference, TREC 2004, Gaithersburg, Maryland, USA, November 16-19, 2004*. Ed. by E. M. Voorhees and L. P. Buckland. Vol. 500-261. *NIST Special Publication*. National Institute of Standards and Technology (NIST). URL: http://trec.nist.gov/pubs/trec13/papers/umass.novelty.hard.pdf.

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

Jégou, H., M. Douze, and C. Schmid. (2011). "Product Quantization for Nearest Neighbor Search". *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 33(1): 117–128. DOI: 10.1109/tpami.2010.57. URL: https://doi.org/10.1109/tpami.2010.57.

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*. ACM. DOI: 10.1145/3269206.3271808. URL: https://doi.org/10.1145/3269206.3271808.

Jiang, Z., F. F. Xu, J. Araki, and G. Neubig. (2020). "How Can We Know What Language Models Know". *Trans. Assoc. Comput. Linguistics*. 8: 423–438. URL: https://transacl.org/ojs/index.php/tacl/article/view/1983.

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 of the Association for Computational Linguistics: EMNLP 2020*. Association for Computational Linguistics. DOI: 10.18653/v1/2020.findings-emnlp.372. URL: https://doi.org/10.18653/v1/2020.findings-emnlp.372.

Johnson, J., A. Karpathy, and L. Fei-Fei. (2016). "DenseCap: Fully Convolutional Localization Networks for Dense Captioning". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE. DOI: 10.1109/cvpr.2016.494. URL: https://doi.org/10.1109/cvpr.2016.494.

Jung, E., J. Choi, and W. Rhee. (2021). “Semi-Siamese Bi-encoder Neural Ranking Model Using Lightweight Fine-Tuning”. *CoRR*. abs/2110.14943. arXiv: 2110.14943. URL: https://arxiv.org/abs/2110.14943.

Kågebäck, M., O. Mogren, N. Tahmasebi, and D. Dubhashi. (2014). “Extractive Summarization using Continuous Vector Space Models”. In: *Proceedings of the 2nd Workshop on Continuous Vector Space Models and their Compositionality (CVSC)*. Association for Computational Linguistics. DOI: 10.3115/v1/w14-1504. URL: https://doi.org/10.3115/v1/w14-1504.

Karpukhin, V., B. Oguz, S. Min, P. Lewis, L. Wu, S. Edunov, D. Chen, and W.-t. Yih. (2020). “Dense Passage Retrieval for Open-Domain Question Answering”. In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics. DOI: 10.18653/v1/2020.emnlp-main.550. URL: https://doi.org/10.18653/v1/2020.emnlp-main.550.

Ke, P., H. Ji, S. Liu, X. Zhu, and M. Huang. (2019). “SentiLR: Linguistic Knowledge Enhanced Language Representation for Sentiment Analysis”. *CoRR*. abs/1911.02493. arXiv: 1911.02493. URL: http://arxiv.org/abs/1911.02493.

Kenter, T. and M. de Rijke. (2015). “Short Text Similarity with Word Embeddings”. In: *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management. CIKM ’15*. Melbourne, Australia: Association for Computing Machinery. 1411–1420. ISBN: 9781450337946. DOI: 10.1145/2806416.2806475. URL: https://doi.org/10.1145/2806416.2806475.

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*. ACM. DOI: 10.1145/3397271.3401075. URL: https://doi.org/10.1145/3397271.3401075.

Kobayashi, H., M. Noguchi, and T. Yatsuka. (2015). “Summarization Based on Embedding Distributions”. In: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics. DOI: 10.18653/v1/d15-1232. URL: https://doi.org/10.18653/v1/d15-1232.

Kulkarni, S., S. Chammas, W. Zhu, F. Sha, and E. Ie. (2020). “AQuaMuSe: Automatically Generating Datasets for Query-Based Multi-Document Summarization”. *CoRR*. abs/2010.12694. arXiv: 2010.12694. URL: https://arxiv.org/abs/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 and Knowledge Management*. ACM. DOI: 10.1145/3340531.3412137. URL: https://doi.org/10.1145/3340531.3412137.

Kurland, O. and L. Lee. (2004). “Corpus structure, language models, and ad hoc information retrieval”. In: *Proceedings of the 27th annual international conference on Research and development in information retrieval - SIGIR ’04*. ACM Press. DOI: 10.1145/1008992.1009027. URL: https://doi.org/10.1145/1008992.1009027.

Kuzi, S., D. Carmel, A. Libov, and A. Raviv. (2017). “Query Expansion for Email Search”. In: *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM. DOI: 10.1145/3077136.3080660. URL: https://doi.org/10.1145/3077136.3080660.

Kuzi, S., A. Shtok, and O. Kurland. (2016). “Query Expansion Using Word Embeddings”. In: *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*. ACM. DOI: 10.1145/2983323.2983876. URL: https://doi.org/10.1145/2983323.2983876.

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”. *CoRR*. abs/2010.01195. arXiv: 2010.01195. URL: https://arxiv.org/abs/2010.01195.

Kwiatkowski, T., J. Palomaki, O. Redfield, M. Collins, A. P. Parikh, C. Alberti, D. Epstein, I. Polosukhin, J. Devlin, K. Lee, K. Toutanova, L. Jones, M. Kelcey, M. Chang, A. M. Dai, J. Uszkoreit, Q. Le, and S. Petrov. (2019). "Natural Questions: a Benchmark for Question Answering Research". *Trans. Assoc. Comput. Linguistics*. 7: 452–466. URL: https://transacl.org/ojs/index.php/tacl/article/view/1455.

Lafferty, J. and C. Zhai. (2003). "Probabilistic relevance models based on document and query generation". In: *Language modeling for information retrieval*. Springer. 1–10.

Lan, Z., M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut. (2020). "ALBERT: A Lite BERT for Self-supervised Learning of Language Representations". In: *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net. URL: https://openreview.net/forum?id=H1eA7AEtvS.

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: *Advances in Artificial Intelligence*. Springer International Publishing. 342–348. DOI: 10.1007/978-3-030-47358-7_35. URL: https://doi.org/10.1007/978-3-030-47358-7_35.

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". In: *Proceedings of the 28th International Conference on Computational Linguistics*. International Committee on Computational Linguistics. DOI: 10.18653/v1/2020.coling-main.495. URL: https://doi.org/10.18653/v1/2020.coling-main.495.

Lavrenko, V. and W. B. Croft. (2017). "Relevance-Based Language Models". *ACM SIGIR Forum*. 51(2): 260–267. DOI: 10.1145/3130348.3130376. URL: https://doi.org/10.1145/3130348.3130376.

Le, Q. V. and T. Mikolov. (2014). “Distributed Representations of Sentences and Documents”. In: *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*. Vol. 32. *JMLR Workshop and Conference Proceedings*. JMLR.org. 1188–1196. URL: http://proceedings.mlr.press/v32/le14.html.

Lee, J., M. Seo, H. Hajishirzi, and J. Kang. (2020). “Contextualized Sparse Representations for Real-Time Open-Domain Question Answering”. In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics. DOI: 10.18653/v1/2020.acl-main.85. URL: https://doi.org/10.18653/v1/2020.acl-main.85.

Lee, J., W. Yoon, S. Kim, D. Kim, S. Kim, C. H. So, and J. Kang. (2019a). “BioBERT: a pre-trained biomedical language representation model for biomedical text mining”. *Bioinformatics*. Sept. Ed. by J. Wren. DOI: 10.1093/bioinformatics/btz682. URL: https://doi.org/10.1093/bioinformatics/btz682.

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*. Association for Computational Linguistics. DOI: 10.18653/v1/p19-1612. URL: https://doi.org/10.18653/v1/p19-1612.

Lee, K.-H., X. Chen, G. Hua, H. Hu, and X. He. (2018). “Stacked Cross Attention for Image-Text Matching”. In: *Computer Vision – ECCV 2018*. Springer International Publishing. 212–228. DOI: 10.1007/978-3-030-01225-0_13. URL: https://doi.org/10.1007/978-3-030-01225-0_13.

Lester, B., R. Al-Rfou, and N. Constant. (2021). “The Power of Scale for Parameter-Efficient Prompt Tuning”. In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics. DOI: 10.18653/v1/2021.emnlp-main.243. URL: https://doi.org/10.18653/v1/2021.emnlp-main.243.

Lewis, M., Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer. (2020a). “BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension”. In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics. DOI: 10.18653/v1/2020.acl-main.703. URL: https://doi.org/10.18653/v1/2020.acl-main.703.

Lewis, P. S. H., E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W. Yih, T. Rocktäschel, S. Riedel, and D. Kiela. (2020b). “Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks”. 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. Balcan, and H. Lin. URL: https://proceedings.neurips.cc/paper/2020/hash/6b493230205f780e1bc26945df7481e5-Abstract.html.

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

Li, H. (2014). “Learning to Rank for Information Retrieval and Natural Language Processing, Second Edition”. *Synthesis Lectures on Human Language Technologies*. 7(3): 1–121. DOI: 10.2200/s00607ed2v01y201410hlt026. URL: https://doi.org/10.2200/s00607ed2v01y201410

Li, H. and J. Xu. (2014). “Semantic Matching in Search”. *Foundations and Trends® in Information Retrieval*. 7(5): 343–469. DOI: 10.1561/1500000035. URL: https://doi.org/10.1561/1500000035.

Li, L. H., M. Yatskar, D. Yin, C.-J. Hsieh, and K.-W. Chang. (2020b). “What Does BERT with Vision Look At?” In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics. DOI: 10.18653/v1/2020.acl-main.469. URL: https://doi.org/10.18653/v1/2020.acl-main.469.

Li, P., C. J. C. Burges, and Q. Wu. (2007). "McRank: Learning to Rank Using Multiple Classification and Gradient Boosting". In: *Advances in Neural Information Processing Systems 20, Proceedings of the Twenty-First Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 3-6, 2007*. Ed. by J. C. Platt, D. Koller, Y. Singer, and S. T. Roweis. Curran Associates, Inc. 897–904. URL: https://proceedings.neurips.cc/paper/2007/hash/b86e8d03fe992d1b0e19656875ee557c-Abstract.html.

Li, W., C. Gao, G. Niu, X. Xiao, H. Liu, J. Liu, H. Wu, and H. Wang. (2021). "UNIMO: Towards Unified-Modal Understanding and Generation via Cross-Modal Contrastive Learning". In: *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Association for Computational Linguistics. DOI: 10.18653/v1/2021.acl-long.202. URL: https://doi.org/10.18653/v1/2021.acl-long.202.

Li, W., Y. Zhang, Y. Sun, W. Wang, M. Li, W. Zhang, and X. Lin. (2020c). "Approximate Nearest Neighbor Search on High Dimensional Data — Experiments, Analyses, and Improvement". *IEEE Transactions on Knowledge and Data Engineering*. 32(8): 1475–1488. DOI: 10.1109/tkde.2019.2909204. URL: https://doi.org/10.1109/tkde.2019.2909204.

Li, X. L. and P. Liang. (2021). "Prefix-Tuning: Optimizing Continuous Prompts for Generation". In: *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Association for Computational Linguistics. DOI: 10.18653/v1/2021.acl-long.353. URL: https://doi.org/10.18653/v1/2021.acl-long.353.

Li, X., X. Yin, C. Li, P. Zhang, X. Hu, L. Zhang, L. Wang, H. Hu, L. Dong, F. Wei, Y. Choi, and J. Gao. (2020d). "Oscar: Object-Semantics Aligned Pre-training for Vision-Language Tasks". In: *Computer Vision – ECCV 2020*. Springer International Publishing. 121–137. DOI: 10.1007/978-3-030-58577-8_8. URL: https://doi.org/10.1007/978-3-030-58577-8_8.

Liang, D., P. Xu, S. Shakeri, C. N. dos Santos, R. Nallapati, Z. Huang, and B. Xiang. (2020). "Embedding-based Zero-shot Retrieval through Query Generation". *CoRR*. abs/2009.10270. arXiv: 2009.10270. URL: https://arxiv.org/abs/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". *CoRR*. abs/2106.14807. arXiv: 2106.14807. URL: https://arxiv.org/abs/2106.14807.

Lin, J., R. Nogueira, and A. Yates. (2021a). "Pretrained Transformers for Text Ranking: BERT and Beyond". *Synthesis Lectures on Human Language Technologies*. 14(4): 1–325. DOI: 10.2200/s01123ed1v01y202108hlt053. URL: https://doi.org/10.2200/s01123ed1v01y202108

Lin, S.-C., J.-H. Yang, and J. Lin. (2021b). "In-Batch Negatives for Knowledge Distillation with Tightly-Coupled Teachers for Dense Retrieval". In: *Proceedings of the 6th Workshop on Representation Learning for NLP (RepL4NLP-2021)*. Association for Computational Linguistics. DOI: 10.18653/v1/2021.repl4nlp-1.17. URL: https://doi.org/10.18653/v1/2021.repl4nlp-1.17.

Lin, S., J. Yang, R. Nogueira, M. Tsai, C. Wang, and J. Lin. (2020). "Query Reformulation using Query History for Passage Retrieval in Conversational Search". *CoRR*. abs/2005.02230. arXiv: 2005.02230. URL: https://arxiv.org/abs/2005.02230.

Liu, B., H. Zamani, X. Lu, and J. S. Culpepper. (2021a). "Generalizing Discriminative Retrieval Models using Generative Tasks". In: *Proceedings of the Web Conference 2021*. ACM. DOI: 10.1145/3442381.3449863. URL: https://doi.org/10.1145/3442381.3449863.

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. DOI: 10.1109/icassp39728.2021.9413557. URL: https://doi.org/10.1109/icassp39728.2021.9413557.

Liu, P., W. Yuan, J. Fu, Z. Jiang, H. Hayashi, and G. Neubig. (2021c). "Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing". *CoRR*. abs/2107.13586. arXiv: 2107.13586. URL: https://arxiv.org/abs/2107.13586.

Liu, R., Z. Lin, and W. Wang. (2021d). "Addressing Extraction and Generation Separately: Keyphrase Prediction With Pre-Trained Language Models". *IEEE/ACM Transactions on Audio, Speech, and Language Processing*. 29: 3180–3191. DOI: 10.1109/taslp.2021.3120587. URL: https://doi.org/10.1109/taslp.2021.3120587.

Liu, T.-Y. (2007). "Learning to Rank for Information Retrieval". *Foundations and Trends® in Information Retrieval*. 3(3): 225–331. DOI: 10.1561/1500000016. URL: https://doi.org/10.1561/1500000016.

Liu, X., F. Zhang, Z. Hou, L. Mian, Z. Wang, J. Zhang, and J. Tang. (2021e). "Self-supervised Learning: Generative or Contrastive". *IEEE Transactions on Knowledge and Data Engineering*: 1–1. DOI: 10.1109/tkde.2021.3090866. URL: https://doi.org/10.1109/tkde.2021.3090866.

Liu, X., Y. Zheng, Z. Du, M. Ding, Y. Qian, Z. Yang, and J. Tang. (2021f). "GPT Understands, Too". *CoRR*. abs/2103.10385. arXiv: 2103.10385. URL: https://arxiv.org/abs/2103.10385.

Liu, X. and W. B. Croft. (2004). "Cluster-based retrieval using language models". In: *Proceedings of the 27th annual international conference on Research and development in information retrieval - SIGIR '04*. ACM Press. DOI: 10.1145/1008992.1009026. URL: https://doi.org/10.1145/1008992.1009026.

Liu, X. and W. B. Croft. (2006). "Statistical language modeling for information retrieval". *Annual Review of Information Science and Technology*. 39(1): 1–31. DOI: 10.1002/aris.1440390108. URL: https://doi.org/10.1002/aris.1440390108.

Liu, Y. and M. Lapata. (2019). "Text Summarization with Pretrained Encoders". In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Association for Computational Linguistics. DOI: 10.18653/v1/d19-1387. URL: https://doi.org/10.18653/v1/d19-1387.

Liu, Y., W. Lu, S. Cheng, D. Shi, S. Wang, Z. Cheng, and D. Yin. (2021g). "Pre-trained Language Model for Web-scale Retrieval in Baidu Search". In: *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. ACM. DOI: 10.1145/3447548.3467149. URL: https://doi.org/10.1145/3447548.3467149.

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". *CoRR*. abs/1907.11692. arXiv: 1907.11692. URL: http://arxiv.org/abs/1907.11692.

Lotze, T., S. Klut, M. Aliannejadi, and E. Kanoulas. (2021). "Ranking Clarifying Questions Based on Predicted User Engagement". *CoRR*. abs/2103.06192. arXiv: 2103.06192. URL: https://arxiv.org/abs/2103.06192.

Lu, J., D. Batra, D. Parikh, and S. Lee. (2019). "ViLBERT: Pretraining Task-Agnostic Visiolinguistic Representations for Vision-and-Language Tasks". In: *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*. Ed. by H. M. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. B. Fox, and R. Garnett. 13–23. URL: https://proceedings.neurips.cc/paper/2019/hash/c74d97b01eae257e44aa9d5bade97baf-Abstract.html.

Lu, S., D. He, C. Xiong, G. Ke, W. Malik, Z. Dou, P. Bennett, T.-Y. Liu, and A. Overwijk. (2021). "Less is More: Pretrain a Strong Siamese Encoder for Dense Text Retrieval Using a Weak Decoder". In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics. DOI: 10.18653/v1/2021.emnlp-main.220. URL: https://doi.org/10.18653/v1/2021.emnlp-main.220.

Luan, Y., J. Eisenstein, K. Toutanova, and M. Collins. (2021). "Sparse, Dense, and Attentional Representations for Text Retrieval". *Transactions of the Association for Computational Linguistics*. 9: 329–345. DOI: 10.1162/tacl_a_00369. URL: https://doi.org/10.1162/tacl_a_00369.

Luo, C., Y. Zheng, Y. Liu, X. Wang, J. Xu, M. Zhang, and S. Ma. (2017a). "SogouT-16". In: *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM. DOI: 10.1145/3077136.3080694. URL: https://doi.org/10.1145/3077136.3080694.

Luo, C., Y. Zheng, J. Mao, Y. Liu, M. Zhang, and S. Ma. (2017b). "Training Deep Ranking Model with Weak Relevance Labels". In: *Lecture Notes in Computer Science*. Springer International Publishing. 205–216. DOI: 10.1007/978-3-319-68155-9_16. URL: https://doi.org/10.1007/978-3-319-68155-9_16.

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*. Association for Computational Linguistics. DOI: 10.18653/v1/2021.eacl-main.92. URL: https://doi.org/10.18653/v1/2021.eacl-main.92.

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". In: *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*. ACM. DOI: 10.1145/3437963.3441777. URL: https://doi.org/10.1145/3437963.3441777.

Ma, X., J. Guo, R. Zhang, Y. Fan, Y. Li, and X. Cheng. (2021c). "B-PROP". In: *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM. DOI: 10.1145/3404835.3462869. URL: https://doi.org/10.1145/3404835.3462869.

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 and Knowledge Management*. ACM. DOI: 10.1145/3459637.3482286. URL: https://doi.org/10.1145/3459637.3482286.

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". In: *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM. DOI: 10.1145/3397271.3401093. URL: https://doi.org/10.1145/3397271.3401093.

MacAvaney, S., A. Yates, A. Cohan, and N. Goharian. (2019). "CEDR: Contextualized Embeddings for Document Ranking". In: *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2019, Paris, France, July 21-25, 2019*. ACM. 1101–1104. DOI: 10.1145/3331184.3331317. URL: https://doi.org/10.1145/3331184.3331317.

Mahata, D., J. Kuriakose, R. R. Shah, and R. Zimmermann. (2018). "Key2Vec: Automatic Ranked Keyphrase Extraction from Scientific Articles using Phrase Embeddings". In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*. Association for Computational Linguistics. DOI: 10.18653/v1/n18-2100. URL: https://doi.org/10.18653/v1/n18-2100.

Mallia, A., O. Khattab, T. Suel, and N. Tonellotto. (2021). "Learning Passage Impacts for Inverted Indexes". In: *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM. DOI: 10.1145/3404835.3463030. URL: https://doi.org/10.1145/3404835.3463030.

Manning, C. D., P. Raghavan, and H. Schütze. (2008). *Introduction to Information Retrieval*. Cambridge University Press. DOI: 10.1017/cbo9780511809071. URL: https://doi.org/10.1017/cbo9780511809071.

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*. Association for Computational Linguistics. DOI: 10.18653/v1/2020.acl-main.74. URL: https://doi.org/10.18653/v1/2020.acl-main.74.

McCann, B., J. Bradbury, C. Xiong, and R. Socher. (2017). "Learned in Translation: Contextualized Word Vectors". In: *Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*. 6294–6305. URL: https://proceedings.neurips.cc/paper/2017/hash/20c86a628232a67e7bd46f76fba7ce12-Abstract.html.

McCreery, C. H., N. Katariya, A. Kannan, M. Chablani, and X. Amatriain. (2020). "Effective Transfer Learning for Identifying Similar Questions". In: *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM. DOI: 10.1145/3394486.3412861. URL: https://doi.org/10.1145/3394486.3412861.

Mei, Q., J. Guo, and D. Radev. (2010). "DivRank". In: *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '10*. ACM Press. DOI: 10.1145/1835804.1835931. URL: https://doi.org/10.1145/1835804.1835931.

Melamud, O., J. Goldberger, and I. Dagan. (2016). "context2vec: Learning Generic Context Embedding with Bidirectional LSTM". In: *CoNLL*.

Metzler, D., Y. Tay, D. Bahri, and M. Najork. (2021). "Rethinking search". *ACM SIGIR Forum*. 55(1): 1–27. DOI: 10.1145/3476415.3476428. URL: https://doi.org/10.1145/3476415.3476428.

Mikolov, T., K. Chen, G. Corrado, and J. Dean. (2013a). "Efficient Estimation of Word Representations in Vector Space". In: *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*. URL: http://arxiv.org/abs/1301.3781.

Mikolov, T., I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. (2013b). "Distributed Representations of Words and Phrases and their Compositionality". In: *27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*. 3111–3119. URL: https://proceedings.neurips.cc/paper/2013/hash/9aa42b31882ec039965f3c4923ce901b-Abstract.html.

Mikolov, T., W. Yih, and G. Zweig. (2013c). “Linguistic Regularities in Continuous Space Word Representations”. In: *Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, June 9-14, 2013, Westin Peachtree Plaza Hotel, Atlanta, Georgia, USA*. The Association for Computational Linguistics. 746–751. URL: https://aclanthology.org/N13-1090/.

Mitra, B. and N. Craswell. (2018). “An Introduction to Neural Information Retrieval t”. *Foundations and Trends® in Information Retrieval*. 13(1): 1–126. DOI: 10.1561/1500000061. URL: https://doi.org/10.1561/1500000061.

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*. International World Wide Web Conferences Steering Committee. DOI: 10.1145/3038912.3052579. URL: https://doi.org/10.1145/3038912.3052579.

Mitra, B., E. T. Nalisnick, N. Craswell, and R. Caruana. (2016). “A Dual Embedding Space Model for Document Ranking”. *CoRR*. abs/1602.01137. arXiv: 1602.01137. URL: http://arxiv.org/abs/1602.01137.

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 and Knowledge Management*. ACM. DOI: 10.1145/3340531.3412067. URL: https://doi.org/10.1145/3340531.3412067.

Mohd, M., R. Jan, and M. Shah. (2020). “Text document summarization using word embedding”. *Expert Systems with Applications*. 143(Apr.): 112958. DOI: 10.1016/j.eswa.2019.112958. URL: https://doi.org/10.1016/j.eswa.2019.112958.

Mustar, A., S. Lamprier, and B. Piwowarski. (2020). “Using BERT and BART for Query Suggestion”. In: *Proceedings of the First Joint Conference of the Information Retrieval Communities in Europe (CIRCLE 2020), Samatan, Gers, France, July 6-9, 2020*. Vol. 2621. *CEUR Workshop Proceedings*. CEUR-WS.org. URL: http://ceur-ws.org/Vol-2621/CIRCLE20%5C_06.pdf.

Naseri, S., J. Dalton, A. Yates, and J. Allan. (2021). “CEQE: Contextualized Embeddings for Query Expansion”. In: *Lecture Notes in Computer Science*. Springer International Publishing. 467–482. DOI: 10.1007/978-3-030-72113-8_31. URL: https://doi.org/10.1007/978-3-030-72113-8_31.

Nema, P., M. M. Khapra, A. Laha, and B. Ravindran. (2017). “Diversity driven attention model for query-based abstractive summarization”. In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics. DOI: 10.18653/v1/p17-1098. URL: https://doi.org/10.18653/v1/p17-1098.

Ni, J., C. Qu, J. Lu, Z. Dai, G. H. Ábrego, J. Ma, V. Y. Zhao, Y. Luan, K. B. Hall, M. Chang, and Y. Yang. (2021a). “Large Dual Encoders Are Generalizable Retrievers”. *CoRR*. abs/2112.07899. arXiv: 2112.07899. URL: https://arxiv.org/abs/2112.07899.

Ni, M., H. Huang, L. Su, E. Cui, T. Bharti, L. Wang, D. Zhang, and N. Duan. (2021b). “M3P: Learning Universal Representations via Multitask Multilingual Multimodal Pre-Training”. In: *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*. Computer Vision Foundation / IEEE. 3977–3986. URL: https://openaccess.thecvf.com/content/CVPR2021/html/Ni%5C_M3P%5C_Learning%5C_Universal%5C_Representations%5C_via%5C_Multitask%5C_Multilingual%5C_Multimodal%5C_Pre-Training%5C_CVPR%5C_2021%5C_paper.html.

Nogueira, R. and K. Cho. (2019). “Passage Re-ranking with BERT”. *CoRR*. abs/1901.04085. arXiv: 1901.04085. URL: http://arxiv.org/abs/1901.04085.

Nogueira, R., Z. Jiang, R. Pradeep, and J. Lin. (2020). “Document Ranking with a Pretrained Sequence-to-Sequence Model”. In: *Findings of the Association for Computational Linguistics: EMNLP 2020*. Association for Computational Linguistics. DOI: 10.18653/v1/2020.findings-emnlp.63. URL: https://doi.org/10.18653/v1/2020.findings-emnlp.63.

Nogueira, R., J. Lin, and A. Epistemic. (2019a). “From doc2query to docTTTTTquery”. *Online preprint*. 6.

Nogueira, R., W. Yang, K. Cho, and J. Lin. (2019b). "Multi-Stage Document Ranking with BERT". *CoRR*. abs/1910.14424. arXiv: 1910.14424. URL: http://arxiv.org/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, A. Angert, E. Banner, V. Khetan, T. McDonnell, A. T. Nguyen, D. Xu, B. C. Wallace, M. de Rijke, and M. Lease. (2017). "Neural information retrieval: at the end of the early years". *Information Retrieval Journal*. 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". In: *Advances in Information Retrieval - 42nd European Conference on IR Research, ECIR 2020, Lisbon, Portugal, April 14-17, 2020, Proceedings, Part II*. Vol. 12036. *Lecture Notes in Computer Science*. Springer. 297–304. DOI: 10.1007/978-3-030-45442-5_37. URL: https://doi.org/10.1007/978-3-030-45442-5%5C_37.

Papagiannopoulou, E. and G. Tsoumakas. (2018). "Local word vectors guiding keyphrase extraction". *Information Processing and Management*. 54(6): 888–902. DOI: 10.1016/j.ipm.2018.06.004. URL: https://doi.org/10.1016/j.ipm.2018.06.004.

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*. International Committee on Computational Linguistics. DOI: 10.18653/v1/2020.coling-main.472. URL: https://doi.org/10.18653/v1/2020.coling-main.472.

Pennington, J., R. Socher, and C. Manning. (2014). "Glove: Global Vectors for Word Representation". In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics. DOI: 10.3115/v1/d14-1162. URL: https://doi.org/10.3115/v1/d14-1162.

Peters, M., M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer. (2018). "Deep Contextualized Word Representations". In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. Association for Computational Linguistics. DOI: 10.18653/v1/n18-1202. URL: https://doi.org/10.18653/v1/n18-1202.

Petroni, F., T. Rocktäschel, S. Riedel, P. Lewis, A. Bakhtin, Y. Wu, and A. Miller. (2019). "Language Models as Knowledge Bases?" In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Association for Computational Linguistics. DOI: 10.18653/v1/d19-1250. URL: https://doi.org/10.18653/v1/d19-1250.

Ponte, J. M. and W. B. Croft. (2017). "A Language Modeling Approach to Information Retrieval". *ACM SIGIR Forum*. 51(2): 202–208. DOI: 10.1145/3130348.3130368. URL: https://doi.org/10.1145/3130348.3130368.

Pradeep, R., R. Nogueira, and J. Lin. (2021). "The Expando-Mono-Duo Design Pattern for Text Ranking with Pretrained Sequence-to-Sequence Models". *CoRR*. abs/2101.05667. arXiv: 2101.05667. URL: https://arxiv.org/abs/2101.05667.

Puri, R. and B. Catanzaro. (2019). "Zero-shot Text Classification With Generative Language Models". *CoRR*. abs/1912.10165. arXiv: 1912.10165. URL: http://arxiv.org/abs/1912.10165.

Qiao, Y., C. Xiong, Z. Liu, and Z. Liu. (2019). "Understanding the Behaviors of BERT in Ranking". *CoRR*. abs/1904.07531. arXiv: 1904.07531. URL: http://arxiv.org/abs/1904.07531.

Qiu, Q., Z. Xie, L. Wu, and W. Li. (2019). "Geoscience keyphrase extraction algorithm using enhanced word embedding". *Expert Systems with Applications*. 125(July): 157–169. DOI: 10.1016/j.eswa.2019.02.001. URL: https://doi.org/10.1016/j.eswa.2019.02.001.

Qiu, X., T. Sun, Y. Xu, Y. Shao, N. Dai, and X. Huang. (2020). "Pre-trained models for natural language processing: A survey". *Science China Technological Sciences*. 63(10): 1872–1897. DOI: 10.1007/s11431-020-1647-3. URL: https://doi.org/10.1007/s11431-020-1647-3.

Qu, Y., Y. Ding, J. Liu, K. Liu, R. Ren, W. X. Zhao, D. Dong, H. Wu, and H. Wang. (2021). "RocketQA: An Optimized Training Approach to Dense Passage Retrieval for Open-Domain Question Answering". In: *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics. DOI: 10.18653/v1/2021.naacl-main.466. URL: https://doi.org/10.18653/v1/2021.naacl-main.466.

Radford, A., K. Narasimhan, T. Salimans, and I. Sutskever. (2018). "Improving language understanding by generative pre-training".

Radford, A., J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, *et al.* (2019). "Language models are unsupervised multitask learners". *OpenAI blog*. 1(8): 9.

Raffel, C., N. 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". *J. Mach. Learn. Res.* 21: 140:1–140:67. URL: http://jmlr.org/papers/v21/20-074.html.

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*.

Reimers, N. and I. Gurevych. (2019). "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks". In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Association for Computational Linguistics. DOI: 10.18653/v1/d19-1410. URL: https://doi.org/10.18653/v1/d19-1410.

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". In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics. DOI: 10.18653/v1/2021.emnlp-main.224. URL: https://doi.org/10.18653/v1/2021.emnlp-main.224.

Robertson, S. E. and K. S. Jones. (1976). "Relevance weighting of search terms". *Journal of the American Society for Information Science*. 27(3): 129–146. DOI: 10.1002/asi.4630270302. URL: https://doi.org/10.1002/asi.4630270302.

Robertson, S. E. and S. Walker. (1994). "Some Simple Effective Approximations to the 2-Poisson Model for Probabilistic Weighted Retrieval". In: *SIGIR '94*. Springer London. 232–241. DOI: 10.1007/978-1-4471-2099-5_24. URL: https://doi.org/10.1007/978-1-4471-2099-5_24.

Robertson, S. and H. Zaragoza. (2009). "The Probabilistic Relevance Framework: BM25 and Beyond". *Foundations and Trends® in Information Retrieval*. 3(4): 333–389. DOI: 10.1561/1500000019. URL: https://doi.org/10.1561/1500000019.

Robertson, S. E., S. Walker, S. Jones, M. Hancock-Beaulieu, and M. Gatford. (1994). "Okapi at TREC-3". In: *Proceedings of The Third Text REtrieval Conference, TREC 1994, Gaithersburg, Maryland, USA, November 2-4, 1994*. Vol. 500-225. *NIST Special Publication*. National Institute of Standards and Technology (NIST). 109–126. URL: http://trec.nist.gov/pubs/trec3/papers/city.ps.gz.

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*. ACM. DOI: 10.1145/3366423.3380009. URL: https://doi.org/10.1145/3366423.3380009.

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*. ACM. DOI: 10.1145/3366423.3380193. URL: https://doi.org/10.1145/3366423.3380193.

Roy, D., D. Ganguly, M. Mitra, and G. J. Jones. (2016a). "Representing documents and queries as sets of word embedded vectors for information retrieval". In: *Proceedings of Neu-IR: The SIGIR 2016 Workshop on Neural Information Retrieval*.

Roy, D., D. Paul, M. Mitra, and U. Garain. (2016b). "Using Word Embeddings for Automatic Query Expansion". *CoRR*. abs/1606.07608. arXiv: 1606.07608. URL: http://arxiv.org/abs/1606.07608.

Sachan, D. S., S. Reddy, W. L. Hamilton, C. Dyer, and D. Yogatama. (2021). "End-to-End Training of Multi-Document Reader and Retriever for Open-Domain Question Answering". *CoRR*. abs/2106.05346. arXiv: 2106.05346. URL: https://arxiv.org/abs/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". In: *Lecture Notes in Computer Science*. Springer International Publishing. 328–335. DOI: 10.1007/978-3-030-45442-5_41. URL: https://doi.org/10.1007/978-3-030-45442-5_41.

Saito, I., K. Nishida, K. Nishida, and J. Tomita. (2020). "Abstractive Summarization with Combination of Pre-trained Sequence-to-Sequence and Saliency Models". *CoRR*. abs/2003.13028. arXiv: 2003.13028. URL: https://arxiv.org/abs/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*. ACM. DOI: 10.1145/3331184.3331326. URL: https://doi.org/10.1145/3331184.3331326.

Salton, G., A. Wong, and C. S. Yang. (1975). "A vector space model for automatic indexing". *Communications of the ACM*. 18(11): 613–620. DOI: 10.1145/361219.361220. URL: https://doi.org/10.1145/361219.361220.

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

Santos, C. N. dos, X. Ma, R. Nallapati, Z. Huang, and B. Xiang. (2020). "Beyond [CLS] through Ranking by Generation". In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics. DOI: 10.18653/v1/2020.emnlp-main.134. URL: https://doi.org/10.18653/v1/2020.emnlp-main.134.

Saracevic, T. (2016). "The Notion of Relevance in Information Science: Everybody knows what relevance is. But, what is it really?" *Synthesis Lectures on Information Concepts, Retrieval, and Services*. 8(3): i–109. DOI: 10.2200/s00723ed1v01y201607icr050. URL: https://doi.org/10.2200/s00723ed1v01y201607icr050.

Savery, M. E., A. B. Abacha, S. Gayen, and D. Demner-Fushman. (2020). "Question-Driven Summarization of Answers to Consumer Health Questions". *CoRR*. abs/2005.09067. arXiv: 2005.09067. URL: https://arxiv.org/abs/2005.09067.

Schick, T. and H. Schütze. (2021a). "Exploiting Cloze-Questions for Few-Shot Text Classification and Natural Language Inference". In: *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*. Association for Computational Linguistics. DOI: 10.18653/v1/2021.eacl-main.20. URL: https://doi.org/10.18653/v1/2021.eacl-main.20.

Schick, T. and H. Schütze. (2021b). "It's Not Just Size That Matters: Small Language Models Are Also Few-Shot Learners". *ArXiv*. abs/2009.07118.

See, A., P. J. Liu, and C. D. Manning. (2017). "Get To The Point: Summarization with Pointer-Generator Networks". In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics. DOI: 10.18653/v1/p17-1099. URL: https://doi.org/10.18653/v1/p17-1099.

Sekulic, I., A. Soleimani, M. Aliannejadi, and F. A. Crestani. (2020). "Longformer for MS MARCO Document Re-ranking Task". *ArXiv*. abs/2009.09392.

Seo, M., J. Lee, T. Kwiatkowski, A. Parikh, A. Farhadi, and H. Hajishirzi. (2019). "Real-Time Open-Domain Question Answering with Dense-Sparse Phrase Index". In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics. DOI: 10.18653/v1/p19-1436. URL: https://doi.org/10.18653/v1/p19-1436.

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*. ACM. DOI: 10.1145/2661829.2661935. URL: https://doi.org/10.1145/2661829.2661935.

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*. ACM. DOI: 10.1145/3077136.3080716. URL: https://doi.org/10.1145/3077136.3080716.

Shi, P., H. Bai, and J. Lin. (2020). "Cross-Lingual Training of Neural Models for Document Ranking". In: *Findings of the Association for Computational Linguistics: EMNLP 2020*. Association for Computational Linguistics. DOI: 10.18653/v1/2020.findings-emnlp.249. URL: https://doi.org/10.18653/v1/2020.findings-emnlp.249.

Shleifer, S. and A. M. Rush. (2020). "Pre-trained Summarization Distillation". *CoRR*. abs/2010.13002. arXiv: 2010.13002. URL: https://arxiv.org/abs/2010.13002.

Singhal, A., C. Buckley, and M. Mitra. (2017). "Pivoted Document Length Normalization". *ACM SIGIR Forum*. 51(2): 176–184. DOI: 10.1145/3130348.3130365. URL: https://doi.org/10.1145/3130348.3130365.

Soldaini, L. and A. Moschitti. (2020). "The Cascade Transformer: an Application for Efficient Answer Sentence Selection". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics. DOI: 10.18653/v1/2020.acl-main.504. URL: https://doi.org/10.18653/v1/2020.acl-main.504.

Song, K., X. Tan, T. Qin, J. Lu, and T. Liu. (2019). “MASS: Masked Sequence to Sequence Pre-training for Language Generation”. In: *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*. Ed. by K. Chaudhuri and R. Salakhutdinov. Vol. 97. *Proceedings of Machine Learning Research*. PMLR. 5926–5936. URL: http://proceedings.mlr.press/v97/song19d.html.

Sordoni, A., Y. Bengio, H. Vahabi, C. Lioma, J. G. 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*. ACM. DOI: 10.1145/2806416.2806493. URL: https://doi.org/10.1145/2806416.2806493.

Su, D., Y. Xu, T. Yu, F. B. Siddique, E. Barezi, and P. Fung. (2020a). “CAiRE-COVID: A Question Answering and Query-focused Multi-Document Summarization System for COVID-19 Scholarly Information Management”. In: *Proceedings of the 1st Workshop on NLP for COVID-19 (Part 2) at EMNLP 2020*. Association for Computational Linguistics. DOI: 10.18653/v1/2020.nlpcovid19-2.14. URL: https://doi.org/10.18653/v1/2020.nlpcovid19-2.14.

Su, W., X. Zhu, Y. Cao, B. Li, L. Lu, F. Wei, and J. Dai. (2020b). “VL-BERT: Pre-training of Generic Visual-Linguistic Representations”. In: *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net. URL: https://openreview.net/forum?id=SygXPaEYvH.

Sun, C., A. Myers, C. Vondrick, K. Murphy, and C. Schmid. (2019a). “VideoBERT: A Joint Model for Video and Language Representation Learning”. In: *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE. DOI: 10.1109/iccv.2019.00756. URL: https://doi.org/10.1109/iccv.2019.00756.

Sun, S., C. Xiong, Z. Liu, Z. Liu, and J. Bao. (2020). “Joint Keyphrase Chunking and Salience Ranking with BERT”. *CoRR*. abs/2004.13639. arXiv: 2004.13639. URL: https://arxiv.org/abs/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". *CoRR*. abs/1904.09223. arXiv: 1904.09223. URL: http://arxiv.org/abs/1904.09223.

Tahami, A. V., K. Ghajar, and A. Shakery. (2020). "Distilling Knowledge for Fast Retrieval-based Chat-bots". In: *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM. DOI: 10.1145/3397271.3401296. URL: https://doi.org/10.1145/3397271.3401296.

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". In: *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Association for Computational Linguistics. DOI: 10.18653/v1/2021.acl-long.392. URL: https://doi.org/10.18653/v1/2021.acl-long.392.

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". *CoRR*. abs/1910.05786. arXiv: 1910.05786. URL: http://arxiv.org/abs/1910.05786.

Tay, Y., V. Q. Tran, M. Dehghani, J. Ni, D. Bahri, H. Mehta, Z. Qin, K. Hui, Z. Zhao, J. P. Gupta, T. Schuster, W. W. Cohen, and D. Metzler. (2022). "Transformer Memory as a Differentiable Search Index". *CoRR*. abs/2202.06991. arXiv: 2202.06991. URL: https://arxiv.org/abs/2202.06991.

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.

Tishby, N. and N. Zaslavsky. (2015). “Deep learning and the information bottleneck principle”. In: *2015 IEEE Information Theory Workshop (ITW)*. IEEE. DOI: 10.1109/itw.2015.7133169. URL: https://doi.org/10.1109/itw.2015.7133169.

Vaswani, A., N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. (2017). “Attention is All you Need”. In: *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*. 5998–6008. URL: https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

Vinyals, O., A. Toshev, S. Bengio, and D. Erhan. (2015). “Show and tell: A neural image caption generator”. In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE. DOI: 10.1109/cvpr.2015.7298935. URL: https://doi.org/10.1109/cvpr.2015.7298935.

Voorhees, E. (2004). “Overview of the TREC 2004 Robust Retrieval Track”. In: *TREC*.

Vulić, I. and M.-F. Moens. (2015). “Monolingual and Cross-Lingual Information Retrieval Models Based on (Bilingual) Word Embeddings”. In: *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval. SIGIR ’15*. Santiago, Chile: Association for Computing Machinery. 363–372. ISBN: 9781450336215. DOI: 10.1145/2766462.2767752. URL: https://doi.org/10.1145/2766462.2767752.

Wang, H., X. Wang, W. Xiong, M. Yu, X. Guo, S. Chang, and W. Y. Wang. (2019). “Self-Supervised Learning for Contextualized Extractive Summarization”. In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics. DOI: 10.18653/v1/p19-1214. URL: https://doi.org/10.18653/v1/p19-1214.

Wang, R., W. Liu, and C. McDonald. (2014). “Corpus-independent generic keyphrase extraction using word embedding vectors”. In: *Software engineering research conference*. Vol. 39. 1–8.

Wang, W., B. Bi, M. Yan, C. Wu, J. Xia, Z. Bao, L. Peng, and L. Si. (2020). "StructBERT: Incorporating Language Structures into Pre-training for Deep Language Understanding". In: *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net. URL: https://openreview.net/forum?id=BJgQ4lSFPH.

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(Mar.): 176–194. DOI: 10.1162/tacl_a_00360. URL: https://doi.org/10.1162/tacl_a_00360.

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". In: *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*. Association for Computational Linguistics. DOI: 10.18653/v1/2021.findings-acl.73. URL: https://doi.org/10.18653/v1/2021.findings-acl.73.

Wu, S. and M. Dredze. (2019). "Beto, Bentz, Becas: The Surprising Cross-Lingual Effectiveness of BERT". In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Association for Computational Linguistics. DOI: 10.18653/v1/d19-1077. URL: https://doi.org/10.18653/v1/d19-1077.

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. Lin. (2020a). "Early Exiting BERT for Efficient Document Ranking". In: *Proceedings of SustaiNLP: Workshop on Simple and Efficient Natural Language Processing*. Association for Computational Linguistics. DOI: 10.18653/v1/2020.sustainlp-1.11. URL: https://doi.org/10.18653/v1/2020.sustainlp-1.11.

Xin, J., R. Tang, J. Lee, Y. Yu, and J. Lin. (2020b). "DeeBERT: Dynamic Early Exiting for Accelerating BERT Inference". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics. DOI: 10.18653/v1/2020.acl-main.204. URL: https://doi.org/10.18653/v1/2020.acl-main.204.

Xin, J., C. Xiong, A. Srinivasan, A. Sharma, D. Jose, and P. N. Bennett. (2021). "Zero-Shot Dense Retrieval with Momentum Adversarial Domain Invariant Representations". *CoRR*. abs/2110.07581. arXiv: 2110.07581. URL: https://arxiv.org/abs/2110.07581.

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*. ACM. DOI: 10.1145/3077136.3080809. URL: https://doi.org/10.1145/3077136.3080809.

Xiong, L., C. Xiong, Y. Li, K. Tang, J. Liu, P. N. Bennett, J. Ahmed, and A. Overwijk. (2021). "Approximate Nearest Neighbor Negative Contrastive Learning for Dense Text Retrieval". In: *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net. URL: https://openreview.net/forum?id=zeFrfgyZln.

Xu, S., X. Zhang, Y. Wu, F. Wei, and M. Zhou. (2020). "Unsupervised Extractive Summarization by Pre-training Hierarchical Transformers". In: *Findings of the Association for Computational Linguistics: EMNLP 2020*. Association for Computational Linguistics. DOI: 10.18653/v1/2020.findings-emnlp.161. URL: https://doi.org/10.18653/v1/2020.findings-emnlp.161.

Yamada, I., A. Asai, and H. Hajishirzi. (2021). “Efficient Passage Retrieval with Hashing for Open-domain Question Answering”. In: *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*. Association for Computational Linguistics. DOI: 10.18653/v1/2021.acl-short.123. URL: https://doi.org/10.18653/v1/2021.acl-short.123.

Yan, M., C. Li, B. Bi, W. Wang, and S. Huang. (2021). “A Unified Pretraining Framework for Passage Ranking and Expansion”. In: *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*. AAAI Press. 4555–4563. URL: https://ojs.aaai.org/index.php/AAAI/article/view/16584.

Yang, L., M. Zhang, C. Li, M. Bendersky, and M. Najork. (2020). “Beyond 512 Tokens: Siamese Multi-depth Transformer-based Hierarchical Encoder for Long-Form Document Matching”. In: *Proceedings of the 29th ACM International Conference on Information and Knowledge Management*. ACM. DOI: 10.1145/3340531.3411908. URL: https://doi.org/10.1145/3340531.3411908.

Yang, Z., Z. Dai, Y. Yang, J. G. Carbonell, R. Salakhutdinov, and Q. V. Le. (2019). “XLNet: Generalized Autoregressive Pretraining for Language Understanding”. In: *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*. Ed. by H. M. Wallach, H. Larochelle, A. Beygelzimer, F. d’Alché-Buc, E. B. Fox, and R. Garnett. 5754–5764. URL: https://proceedings.neurips.cc/paper/2019/hash/dc6a7e655d7e5840e66733e9ee67cc69-Abstract.html.

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”. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM. DOI: 10.1145/2939672.2939677. URL: https://doi.org/10.1145/2939672.2939677.

Yin, W. and Y. Pei. (2015). "Optimizing Sentence Modeling and Selection for Document Summarization". In: *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015*. AAAI Press. 1383–1389. URL: http://ijcai.org/Abstract/15/199.

Yu, S., Z. Liu, C. Xiong, T. Feng, and Z. Liu. (2021). "Few-Shot Conversational Dense Retrieval". In: *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM. DOI: 10.1145/3404835.3462856. URL: https://doi.org/10.1145/3404835.3462856.

Zamani, H. and W. B. Croft. (2016). "Embedding-based Query Language Models". In: *Proceedings of the 2016 ACM International Conference on the Theory of Information Retrieval*. ACM. DOI: 10.1145/2970398.2970405. URL: https://doi.org/10.1145/2970398.2970405.

Zamani, H. and W. B. Croft. (2017). "Relevance-based Word Embedding". In: *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM. DOI: 10.1145/3077136.3080831. URL: https://doi.org/10.1145/3077136.3080831.

Zamani, H. and W. B. Croft. (2018). "On the Theory of Weak Supervision for Information Retrieval". In: *Proceedings of the 2018 ACM SIGIR International Conference on Theory of Information Retrieval*. ACM. DOI: 10.1145/3234944.3234968. URL: https://doi.org/10.1145/3234944.3234968.

Zamani, H., W. B. Croft, and J. S. Culpepper. (2018a). "Neural Query Performance Prediction using Weak Supervision from Multiple Signals". In: *The 41st International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM. DOI: 10.1145/3209978.3210041. URL: https://doi.org/10.1145/3209978.3210041.

Zamani, H., M. Dehghani, W. B. Croft, E. Learned-Miller, and J. Kamps. (2018b). "From Neural Re-Ranking to Neural Ranking: Learning a Sparse Representation for Inverted Indexing". In: *Proceedings of the 27th ACM International Conference on Information and Knowledge Management. CIKM '18*. Torino, Italy: Association for Computing Machinery. 497–506. ISBN: 9781450360142. DOI: 10.1145/3269206.3271800. URL: https://doi.org/10.1145/3269206.3271800.

Zhai, C. (2007). "Statistical Language Models for Information Retrieval A Critical Review". *Foundations and Trends® in Information Retrieval*. 2(3): 137–213. DOI: 10.1561/1500000008. URL: https://doi.org/10.1561/1500000008.

Zhai, C. and J. Lafferty. (2004). "A study of smoothing methods for language models applied to information retrieval". *ACM Transactions on Information Systems*. 22(2): 179–214. DOI: 10.1145/984321.984322. URL: https://doi.org/10.1145/984321.984322.

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". In: *Proceedings of the 30th ACM International Conference on Information and Knowledge Management*. ACM. DOI: 10.1145/3459637.3482358. URL: https://doi.org/10.1145/3459637.3482358.

Zhan, J., J. Mao, Y. Liu, J. Guo, M. Zhang, and S. Ma. (2021b). "Optimizing Dense Retrieval Model Training with Hard Negatives". In: *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM. DOI: 10.1145/3404835.3462880. URL: https://doi.org/10.1145/3404835.3462880.

Zhan, J., J. Mao, Y. Liu, J. Guo, M. Zhang, and S. Ma. (2022). "Learning Discrete Representations via Constrained Clustering for Effective and Efficient Dense Retrieval". In: *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*. ACM. DOI: 10.1145/3488560.3498443. URL: https://doi.org/10.1145/3488560.3498443.

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". *CoRR*. abs/2010.10469. arXiv: 2010.10469. URL: https://arxiv.org/abs/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. (2021a). "Joint Learning of Deep Retrieval Model and Product Quantization based Embedding Index". In: *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM. DOI: 10.1145/3404835.3462988. URL: https://doi.org/10.1145/3404835.3462988.

Zhang, H., H. Shen, Y. Qiu, Y. Jiang, S. Wang, S. Xu, Y. Xiao, B. Long, and W.-Y. Yang. (2021b). "Joint Learning of Deep Retrieval Model and Product Quantization based Embedding Index". In: *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM. DOI: 10.1145/3404835.3462988. URL: https://doi.org/10.1145/3404835.3462988.

Zhang, H., J. Cai, J. Xu, and J. Wang. (2019a). "Pretraining-Based Natural Language Generation for Text Summarization". In: *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*. Association for Computational Linguistics. DOI: 10.18653/v1/k19-1074. URL: https://doi.org/10.18653/v1/k19-1074.

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

Zhang, K., C. Xiong, Z. Liu, and Z. Liu. (2020b). "Selective Weak Supervision for Neural Information Retrieval". In: *Proceedings of The Web Conference 2020*. ACM. DOI: 10.1145/3366423.3380131. URL: https://doi.org/10.1145/3366423.3380131.

Zhang, X., F. Wei, and M. Zhou. (2019b). “HIBERT: Document Level Pre-training of Hierarchical Bidirectional Transformers for Document Summarization”. In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics. DOI: 10.18653/v1/p19-1499. URL: https://doi.org/10.18653/v1/p19-1499.

Zhang, Z., X. Han, Z. Liu, X. Jiang, M. Sun, and Q. Liu. (2019c). “ERNIE: Enhanced Language Representation with Informative Entities”. In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics. DOI: 10.18653/v1/p19-1139. URL: https://doi.org/10.18653/v1/p19-1139.

Zheng, G. and J. Callan. (2015). “Learning to Reweight Terms with Distributed Representations”. In: *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval. SIGIR ’15*. Santiago, Chile: Association for Computing Machinery. 575–584. ISBN: 9781450336215. DOI: 10.1145/2766462.2767700. URL: https://doi.org/10.1145/2766462.2767700.

Zheng, Z., K. Hui, B. He, X. Han, L. Sun, and A. Yates. (2020). “BERT-QE: Contextualized Query Expansion for Document Re-ranking”. In: *Findings of the Association for Computational Linguistics: EMNLP 2020*. Association for Computational Linguistics. DOI: 10.18653/v1/2020.findings-emnlp.424. URL: https://doi.org/10.18653/v1/2020.findings-emnlp.424.

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 and Management*. 58(5): 102672. DOI: 10.1016/j.ipm.2021.102672. URL: https://doi.org/10.1016/j.ipm.2021.102672.

Zhong, M., P. Liu, Y. Chen, D. Wang, X. Qiu, and X. Huang. (2020). “Extractive Summarization as Text Matching”. In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics. DOI: 10.18653/v1/2020.acl-main.552. URL: https://doi.org/10.18653/v1/2020.acl-main.552.

Zhong, M., P. Liu, D. Wang, X. Qiu, and X. Huang. (2019). "Searching for Effective Neural Extractive Summarization: What Works and What's Next". In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics. DOI: 10.18653/v1/p19-1100. URL: https://doi.org/10.18653/v1/p19-1100.

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*. ACM. DOI: 10.1145/3404835.3462918. URL: https://doi.org/10.1145/3404835.3462918.

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*. ACM. DOI: 10.1145/3397271.3401175. URL: https://doi.org/10.1145/3397271.3401175.

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 and Knowledge Management*. ACM. DOI: 10.1145/3459637.3482379. URL: https://doi.org/10.1145/3459637.3482379.

Zhou, Y., J. Yao, Z. Dou, L. Wu, and J. Wen. (2022). "DynamicRetriever: A Pre-training Model-based IR System with Neither Sparse nor Dense Index". *CoRR*. abs/2203.00537. DOI: 10.48550/arXiv.2203.00537. arXiv: 2203.00537. URL: https://doi.org/10.48550/arXiv.2203.00537.

Zhu, H., L. Dong, F. Wei, B. Qin, and T. Liu. (2019). "Transforming Wikipedia into Augmented Data for Query-Focused Summarization". *CoRR*. abs/1911.03324. arXiv: 1911.03324. URL: http://arxiv.org/abs/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 and Knowledge Management*. ACM. DOI: 10.1145/3459637.3482243. URL: https://doi.org/10.1145/3459637.3482243.

Zou, Y., X. Zhang, W. Lu, F. Wei, and M. Zhou. (2020). "Pre-training for Abstractive Document Summarization by Reinstating Source Text". In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics. DOI: 10.18653/v1/2020.emnlp-main.297. URL: https://doi.org/10.18653/v1/2020.emnlp-main.297.

Zuccon, G., B. Koopman, P. Bruza, and L. Azzopardi. (2015). "Integrating and Evaluating Neural Word Embeddings in Information Retrieval". In: *Proceedings of the 20th Australasian Document Computing Symposium*. ACM. DOI: 10.1145/2838931.2838936. URL: https://doi.org/10.1145/2838931.2838936.