A SUPPLEMENTARY CONTENT

In this section, we introduce the available datasets, the evaluation metrics, and the supporting libraries for dense retrieval.

Table 4. The detailed statistics of available retrieval datasets. Here, "q" is the abbreviation of queries, "instance" denotes a candidate text in the collection.

Categorization	Domain	Dataset	#q in train	#label in train	#q in dev	#q in test	#instance
Information retrieval	Web	MS MARCO [180]	502,939	532,761	6,980	6,837	8,841,823
	Web	mMARCO [20]	808,731	–	101,093	–	8,841,823
	News	TREC-NEWS [234]	–	–	–	57	594,977
	Biomedical	TREC-COVID [213]	–	–	–	50	171,332
	Biomedical	NFCorpus [22]	5,922	110,575	324	323	3,633
	Twitter	Signal-1M [238]	–	–	–	97	2,866,316
	Argument	Touché-2020 [19]	–	–	–	249	528,155
	Argument	ArguAna [259]	–	–	–	1,406	8,674
	Wikipedia	DBPedia [81]	–	–	67	400	4,635,922
	Web	ORCAS [44]	10.4M	18.8M	–	–	3,213,835
	Wikipedia	EntityQuestions [225]	176,560	186,367	22,068	22,075	–
	Web	MS MARCO v2 [46]	277,144	284,212	8,184	–	138,364,198
	Web	$\text{DuReader}_{\text{retrieval}}$ [198]	97,343	86,395	2,000	8,948	8,096,668
Question answering	Wikipedia	Natural Questions [114]	152,148	152,148	6,515	3,610	2,681,468
	Wikipedia	SQuAD [203]	78,713	78,713	8,886	10,570	23,215
	Wikipedia	TriviaQA [106]	78,785	78,785	8,837	11,313	740K
	Wikipedia	HotpotQA [284]	85,000	170,000	5,447	7,405	5,233,329
	Web	WebQuestions [15]	3,417	3,417	361	2,032	–
	Web	CuratedTREC [12]	1,353	1,353	133	694	–
	Finance	FiQA-2018 [170]	5,500	14,166	500	648	57,638
	Biomedical	BioASQ [253]	3,743	35,285	–	497	15,559,157
	StackEx.	CQADupStack [93]	–	–	–	13,145	457,199
	Quora	Quora [98]	–	–	5,000	10,000	522,931
	News	ArchivalQA [261]	853,644	853,644	106,706	106,706	483,604
	Web	CCQA [96]	55M	130M	–	–	–
Other tasks	Wikipedia	FEVER [249]	–	140,085	6,666	6,666	5,416,568
	Wikipedia	Climate-FEVER [53]	–	–	–	1,535	5,416,593
	Scitific	SciFact [260]	809	920	–	300	5,183
	Scitific	SciDocs [42]	–	–	–	1,000	25,657

A.1 Datasets

Compared with traditional retrieval models, dense retrieval models are more data hungry, requiring large-scale labeled datasets to learn the parameters of the PLMs. In recent years, there are a number of retrieval datasets with relevance judgements released publicly, which significantly advances the research of dense retrieval. We categorize the available retrieval datasets into three major categories according to their original tasks, namely information retrieval, question answering and other task settings. The statistics of the available datasets are shown in Table 4.

As we can see from Table 4, Wikipedia and Web are two major resources for creating these datasets. Among these datasets, *MS MARCO* dataset [180] contains a large amount of queries with annotated relevant passages in Web documents, and *Natural Questions (NQ)* [114] dataset contains Google search queries and documents with paragraphs and answer spans from the top ranked Wikipedia pages. Among these datasets, MS MARCO and NQ datasets have been widely used for evaluating dense retrieval models. A recent study [47] has summarized the promotion effect of MS MARCO on the progress of dense retrieval: neural models can explore large-scale data for training. Besides, based on MS MARCO, several variants have been created in order to enrich the evaluation characteristics on some specific aspect, *e.g.,* the multilingual version mMARCO [20] and the MS MARCO Chameleons dataset [3] (consisting of obstinate queries that are difficult to answer by neural retrievers and have similar query length and distribution of relevance judgements with easier queries). Besides, Sciavolino et al. [225] create EntityQuestions dataset, as a challenging test set for the models trained on NQ, which contains simple factoid questions about entities from

Wikipedia. Another interesting observation is that there are increasingly more domain-specific retrieval datasets, including COVID-19 pandemic dataset [213], financial dataset [170], biomedical dataset [253], climate-specific dataset [53] and scientific dataset [260].

Besides the presented datasets in Table 4, several more comprehensive benchmark datasets are released to evaluate the overall retrieval capability of the retrieval models by aggregating representative datasets and conducting diverse evaluation tasks, such as *BEIR* [248] and *KILT* [191].

Although existing datasets largely improve the training of dense retrievers, in these datasets, a query typically corresponds to very few relevance judgements. For example, Nogueira et al. observe that most of queries in MS MARCO dataset contains only one labeled positive[4], which is likely to be smaller than the actual number of relevant ones in the collection. It is mainly because it is time-consuming to construct complete relevance judgements for a large dataset. The incomplete relevance annotations will lead to several training issues such as false negative, which potentially affects the retrieval performance.

Besides, to date, most of the released datasets are created in English, and it is more difficult to obtain sufficient labeled data for training a non-English dense retriever. More recently, DuReader-retrieval [198] releases a large-scale Chinese dataset consisting of 90K queries from Baidu search and over 8M passages for passage retrieval. In order to enhance the evaluation quality, DuReader-retrieval tries to reduce the false negatives in development and testing sets, and also removes the training queries that are semantically similar to the development and testing queries. Besides, DuReader-retrieval provides human-translated queries (in English) for cross-lingual retrieval.

A.2 Evaluation Metrics

To evaluate the retrieval capacity of an information retrieval system, a number of factors need to be considered [80, 80, 255]: effectiveness, efficiency, diversity, novelty and so on. This survey mainly focuses on the effectiveness for the retrieval system[20]. We next introduce the commonly used evaluation metrics for ranking, including Recall, Precision, MAP, MRR and NDCG. For dense retrieval tasks, top ranked texts are more important for evaluation, and therefore cut-off metrics are often adopted to examine the quality of texts at top positions. Next, we introduce several commonly used metrics for dense retrieval in detail.

In the traditional retrieval benchmarks, Recall is the fraction of relevant texts that are actually retrieved by a retrieval model among all the relevant ones, and Recall@k [308] calculates a truncated Recall value at the k-th position of a retrieved list:

$$\text{Recall@}k = \frac{1}{|Q|} \sum_{q=1}^{|Q|} \frac{\#\text{retr}_{q,k}}{\#\text{rel}_q}, \tag{7}$$

where $\#\text{retr}_{q,k}$ denotes the number of relevant texts *w.r.t.* query q retrieved at top k positions by a retrieval method, and $\#\text{rel}_q$ denotes the total number of relevant texts for query q. Here, we average the Recall@k values over the queries from the query set Q.

In dense retrieval, there is a commonly used metric, *Top-k Accuracy*, and it computes the proportion of queries for which the top-k retrieved texts contain the answers [108], defined as:

$$\text{Accuracy@}k = \frac{1}{|Q|} \sum_{q=1}^{|Q|} \mathbb{I}(\#\text{retr}_{q,k} > 0), \tag{8}$$

where $\mathbb{I}(\cdot)$ is an binary indicator function that only returns 1 when the case is true. In contrast to traditional TREC benchmarks, mainstream dense retrieval benchmarks, such as NQ, aim to find

[20]Note that these metrics can be used in both first-stage retrieval and reranking. However, we only describe the evaluation metrics from a general ranking perspective without considering specific task scenarios.

answers to queries instead of retrieving all relevant texts. According to [108, 199], a retrieved list is considered to *accurately* solve a query when it contains the answer, not necessarily retrieving all the relevant texts [21].

Besides, Precision@k [308] calculates the average proportion of relevant texts among the top k positions over the query set Q:

$$\text{Precision@}k = \frac{1}{|Q|}\sum_{q=1}^{|Q|}\frac{\#\text{retr}_{q,k}}{k}. \tag{9}$$

Based on Precision@k, the Average Precision (AP) further averages the precision values at the positions of each positive text for a query:

$$\text{AP}_q = \frac{1}{\#\text{rel}_q}\sum_{k=1}^{L}\text{Precision@}l \times \mathbb{I}(q,k), \tag{10}$$

where Precision@l is the per-query version of the precision value at the l-th position, L is the length of a retrieved list and $\mathbb{I}(q,l)$ is an indicator function returning 1 only when the l-th position corresponds to a relevant text for query q. Furthermore, Mean Average Precision (MAP) [308] calculates the average Precision scores over a set of queries Q:

$$\text{MAP} = \frac{1}{|Q|}\sum_{q=1}^{|Q|}\text{AP}_q. \tag{11}$$

Not only counting the occurrence of positive texts, DCG (Normalized Discounted Cumulative Gain) [101] further incorporates the position of a relevant text into consideration: it prefers a ranking that places a relevant text at a higher position:

$$\text{DCG}_q@k = \sum_{i=1}^{k}\frac{2^{g_i}-1}{\log_2(i+1)}, \tag{12}$$

where g_i is the graded relevance score for the i-th retrieved text. Based on the above definition of DCG, NDCG is the sum of the normalized DCG values at a particular rank position:

$$\text{nDCG@}k = \frac{1}{|Q|}\sum_{q=1}^{|Q|}\frac{\text{DCG}_q@k}{\text{IDCG}_q@k}, \tag{13}$$

where DCG@k and IDCG@k denote discounted cumulative gain and ideal discounted cumulative gain at a particular rank position k, respectively.

Furthermore, MRR [258] averages the reciprocal of the rank of the first retrieved positive text over a set of queries Q:

$$\text{MRR} = \frac{1}{|Q|}\sum_{q=1}^{|Q|}\frac{1}{\text{rank}_q}, \tag{14}$$

where rank_q is the position of the first retrieved positive text *w.r.t.* query q.

For first-stage retrieval, Recall@k and Accuracy@k are the most commonly used metrics, since its main focus is to recall as many relevant texts or answers as possible at a truncation length; while for ranking, MRR, NDCG and MAP are more commonly used in practice. For a comprehensive discussion about IR evaluation, the readers are suggested to read more focused references [80, 172, 222, 255].

[21]Note that Accuracy@k is also known as Recall@k in some dense retrieval studies [65, 199], which is somehow different from the definition used in traditional retrieval benchmarks.

A.3 Code Library Resource

Recently, several open-sourced dense retrieval libraries have been released for research purpose. As a representative library, Tevatron [68] has developed a modularized framework for building dense retrieval models based on PLMs via command-line interfaces. It supports a number of important procedures involved in a complete retrieval pipeline including text processing, model training, text encoding, and text retrieval. It can be accessed at the link: https://github.com/TextTron/Tevatron.

Besides, Pyserini [131] is a toolkit that is designed to facilitate reproducible research for information retrieval. Specifically, it supports both sparse retrieval and dense retrieval with Anserini IR toolkit [281] and FAISS [105]. It also provides the evaluation scripts for the standard IR test collections. It can be accessed from the link: http://pyserini.io/.

SentenceTransformers [206] is another library that provides an easy way to compute dense embeddings for sentences and paragraphs based on Transformer-based networks. Specifically, it integrates the implementation of Sentence-BERT [206] and Transformer-based Sequential Denoising Auto-Encoder (TSDAE) [263]. It can be accessed at the link: https://www.sbert.net/.

In order to enhance the validation of dense retriever checkpoints, Asyncval [312] is released to ease and accelerate the checkpoint validation for dense retrieval models. An important merit is that the training can be decoupled from checkpoint validation with Asyncval. It can be accessible at the link: https://github.com/ielab/asyncval.

OpenMatch [152] has been originally proposed for neural information retrieval (v1), and extended (v2) to support dense retrieval on commonly used benchmarks such as MS MARCO and NQ. It can be accessed at the link: https://github.com/thunlp/OpenMatch.

MatchZoo [76] is a text matching library that supports a number of neural text matching models, and allow users to develop new models by providing rich interfaces. It can be accessed at the link: https://github.com/NTMC-Community/MatchZoo.

As the supporting resource, we also release an open-sourced implementation of dense retrievers based our previous work RocketQA [199], at the link: https://github.com/PaddlePaddle/RocketQA, including RocketQA [199], RocketQA$_{PAIR}$ [209] and RocketQAv2 [210]. This software also provides an easy-to-use toolkit with pre-built models (including both English and Chinese models) for direct use after the installation. We also aggregate other open-sourced codes for related dense retrieval papers in our survey site, which can be found in the link: https://github.com/RUCAIBox/DenseRetrieval.