



# Evaluating Interpolation and Extrapolation Performance of Neural Retrieval Models

Jingtao Zhan  
BNRist, DCST, Tsinghua University  
Beijing, China  
jingtaozhan@gmail.com

Xiaohui Xie  
BNRist, DCST, Tsinghua University  
Beijing, China  
xiexiaohui@mail.tsinghua.edu.cn

Jiaxin Mao  
Gaoling School of Artificial  
Intelligence, Renmin University of  
China  
Beijing, China  
maojiaxin@gmail.com

Yiqun Liu\*  
BNRist, DCST, Tsinghua University  
Beijing, China  
yiqunliu@tsinghua.edu.cn

Jiafeng Guo  
CAS Key Lab of Network Data  
Science and Technology, ICT, CAS  
University of Chinese Academy of  
Sciences  
Beijing, China  
guojiafeng@ict.ac.cn

Min Zhang  
Shaoping Ma  
BNRist, DCST, Tsinghua University  
Beijing, China  
z-m@tsinghua.edu.cn  
msp@tsinghua.edu.cn

## ABSTRACT

A retrieval model should not only *interpolate* the training data but also *extrapolate* well to the queries that are different from the training data. While neural retrieval models have demonstrated impressive performance on ad-hoc search benchmarks, we still know little about how they perform in terms of interpolation and extrapolation. In this paper, we demonstrate the importance of separately evaluating the two capabilities of neural retrieval models. Firstly, we examine existing ad-hoc search benchmarks from the two perspectives. We investigate the distribution of training and test data and find a considerable overlap in query entities, query intent, and relevance labels. This finding implies that the evaluation on these test sets is biased toward interpolation and cannot accurately reflect the extrapolation capacity. Secondly, we propose a novel evaluation protocol to separately evaluate the interpolation and extrapolation performance on existing benchmark datasets. It resamples the training and test data based on query similarity and utilizes the resampled dataset for training and evaluation. Finally, we leverage the proposed evaluation protocol to comprehensively revisit a number of widely-adopted neural retrieval models. Results show models perform differently when moving from interpolation to extrapolation. For example, representation-based retrieval models perform almost as well as interaction-based retrieval models in terms of interpolation but not extrapolation. Therefore, it is

necessary to separately evaluate both interpolation and extrapolation performance and the proposed resampling method serves as a simple yet effective evaluation tool for future IR studies.

## CCS CONCEPTS

• **Information systems** → **Evaluation of retrieval results; Retrieval models and ranking.**

## KEYWORDS

evaluation, extrapolation, interpolation, neural ranking

## ACM Reference Format:

Jingtao Zhan, Xiaohui Xie, Jiaxin Mao, Yiqun Liu, Jiafeng Guo, Min Zhang, and Shaoping Ma. 2022. Evaluating Interpolation and Extrapolation Performance of Neural Retrieval Models. In *Proceedings of the 31st ACM International Conference on Information and Knowledge Management (CIKM '22)*, October 17–21, 2022, Atlanta, GA, USA. ACM, New York, NY, USA, 11 pages. <https://doi.org/10.1145/3511808.3557312>

## 1 INTRODUCTION

Ranking is essential for many IR-related tasks, such as ad-hoc search [51, 55], Web search [30], and open domain question answering [19, 23]. Recently, there has been a surge of research interest in applying neural models for first-stage retrieval to improve ranking performance considering their strong capacity for semantic matching. Various model architectures [16, 23, 24, 27, 34] and training techniques [18, 21, 29, 32, 38, 51, 55] are proposed. They are typically evaluated and compared on ad-hoc retrieval benchmarks such as MS MARCO [3] and TREC Deep Learning Tracks [10, 12]. The evaluation protocol is to train the models with massive annotation labels and then evaluate them on a held-out test set. Results suggest that neural retrieval models strongly outperform traditional term-based retrieval methods like BM25 [39] and drive test-set performance to new heights [18, 23, 38, 56].

However, an overall evaluation metric score on a dataset with unclear training-test distribution does not show the full picture of model performance. In this paper, we break the model capacity

\*Corresponding author

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

CIKM '22, October 17–21, 2022, Atlanta, GA, USA

© 2022 Association for Computing Machinery.

ACM ISBN 978-1-4503-9236-5/22/10...\$15.00

<https://doi.org/10.1145/3511808.3557312>

into interpolation and extrapolation regimes, which are two fundamental concepts in machine learning researches [4, 6, 20, 49, 52]. Because queries play a central role in ad-hoc search, we *define the interpolation performance for ranking as how ranking models handle test queries that are similar to training queries*. Similarly, *the extrapolation performance is defined as how ranking models perform on test queries that are distinct from training queries*. It is commonly believed that extrapolation is a more intricate task than interpolation [4, 40]. It is also a more important task for neural retrieval models since retrieval systems should be able to handle arbitrary search requests. Nevertheless, despite the impressive benchmark performance scores of neural retrieval models, we know relatively little about how they perform in interpolation and extrapolation scenarios, respectively.

We first analyze and interpret test-set scores on the benchmarks from the interpolation and extrapolation perspectives in Section 3. We investigate the distribution of training and test data on TREC Deep Learning Tracks [10, 12] and MS MARCO dataset [3], both of which are popular ranking benchmarks [11]. The investigation leads to an alarming finding that a substantial training-test overlap exists in terms of query entities, query intents, and relevance labels. Therefore, the benchmarks largely evaluate the interpolation ability but fail to accurately reflect the extrapolation ability.

Second, we propose a new evaluation protocol to evaluate the interpolation and extrapolation performance based on an existing Cranfield-like dataset in Section 4. We encode the queries in an embedding space and use embedding distance as query similarity. Then we resample similar or dissimilar training-test data to construct a new dataset for the evaluation of interpolation or extrapolation performance, respectively. Models are trained and evaluated on the new dataset. The proposed evaluation method does not require any new test datasets or relevance annotation. To investigate whether it is limited to the distribution that the original dataset follows, we investigate its alignment with generalization performance evaluated by a new out-of-distribution test set. Results suggest that extrapolation performance strongly correlates to out-of-distribution performance while interpolation correlates poorly. Therefore, the proposed evaluation protocol also indicates generalization performance beyond the dataset used for resampling.

Finally, we utilize the proposed evaluation protocol to comprehensively revisit various neural retrieval models in Section 5. We study whether the evaluation results in terms of interpolation and extrapolation lead to different outcomes in model comparisons. We evaluate different model architectures including interaction-based models (ColBERT [24]) and representation-based models (dense retrieval [23] and neural sparse retrieval [16]). We also evaluate some popular training techniques including hard negatives [51, 55], distillation [21, 29], and pretraining [17, 18]. Results demonstrate a substantial disparity in model effectiveness evaluated in the two regimes. For example, interaction-based models extrapolate well, whereas dense retrieval and neural sparse retrieval suffer from severe effectiveness degeneration when moving from interpolation to extrapolation. Besides, hard negative mining and distillation hardly mitigate the performance gap between the two regimes, while the gap is tightly related to the pretraining technique. Therefore, it is

necessary to evaluate the interpolation and extrapolation capacities separately to gain a deep understanding of model performance<sup>1</sup>.

## 2 INTERPOLATION AND EXTRAPOLATION

Interpolation and extrapolation are fundamental concepts in characterizing the behavior of machine learning algorithms on different test cases. While there is no widely-accepted precise formulation, an intuitive and informal definition is commonly agreed [5, 6, 49]: interpolation occurs when the training set is fully representative of the test data, while extrapolation occurs when the training data differs distinctly from the test data. For example, if a machine-learning algorithm is required to apply arithmetics to much larger numbers than those seen during training, it has to extrapolate. Humans extrapolate well in many scenarios because the learning process is beyond simple pattern recognition and the learned knowledge is generic [26]. On the contrary, machine learning algorithms excel at interpolation but may be vulnerable to extrapolation [6, 20]. To achieve high extrapolation accuracy, the algorithms have to learn faithful knowledge that generally holds for all possible inputs instead of spurious correlations that only work for most training data. Evaluating the extrapolation performance is of utmost importance to building trustworthy and robust systems that can be applied in the real world. It has fostered a bunch of machine-learning research in exploring generalizable neural networks [6, 20, 52] and designing robust optimization algorithms [25, 41, 49].

Although ranking is an application area of machine learning, few studies consider interpolation and extrapolation when evaluating ranking models. Since the Cranfield experiments [8], evaluation studies in IR focus on measuring the effectiveness of search systems on a test collection but largely overlook the training process of ranking models and the relationship between training and test data. A typical IR test collection consists of three major components: a corpus of texts, a set of test queries, and a set of relevance labels for some query-text pairs. To evaluate the effectiveness of a ranking model, we can compute some evaluation metrics for each test query with the relevance labels. Based on different assumptions about user behavior, plenty of effort has been put to propose evaluation metrics that reflect user satisfaction, such as NDCG [22], RBP [36], INST [2], BPM [57], IFT [1], etc. While such test-collection-based evaluations have been widely adopted in evaluating ranking models, most existing test collections do not contain any training data. With the emergence of neural IR models, some recent test collections, such as MS MARCO [3], TREC Deep Learning Tracks [10, 12], and NTCIR-WWW2&3 [35, 42], provide large-scale training sets to facilitate the employment of neural ranking models. However, few researchers and practitioners have considered how the training data may impact the performance on test queries, which is especially important in the large data regime where the superior performance of neural ranking models originates from massive training queries.

In this paper, we argue that interpolation and extrapolation are two important perspectives for a proper evaluation of the ranking models that need to be trained with large-scale training data. A model with strong interpolation performance does not necessarily extrapolate well. If the ranking models learn misleading heuristics about relevance that are only suitable for training data, they may

<sup>1</sup>Data is available at <https://github.com/jingtaozhan/extrapolate-eval>.

still interpolate well on similar inputs but extrapolate substantially worse. Therefore, high metric scores on interpolation test collections may not indicate high user satisfaction when the models are applied in the real world where extrapolation is the norm.

Now we consider how to define interpolation and extrapolation in IR. Note that machine-learning researchers usually agree on an intuitive and informal definition that interpolation and extrapolation are the cases where test points are similar to or considerably different from the training data, respectively [5, 6, 49]. As for precise definitions, there are ongoing arguments about what definitions better align with the generalization performance<sup>2</sup>. In this paper, we do not address the problem of mathematically formalizing the two concepts. Instead, we explore a practical definition that characterizes the generalization performance of ranking models when deployed in real-world search engines. In the context of ad-hoc search, queries are of utmost importance to both users and ranking systems. Queries are abstractions of the user intent, and the ranking systems rely on queries to effectively retrieve relevant items that satisfy the users. Queries are not static but dynamically changing: query formulations differ among users, user intents vary across domains, and user interest shifts over time. An effective ranking system should be able to deal with any possible query no matter whether it is similar to the training queries or not. Therefore, we propose to define interpolation and extrapolation in ad-hoc search as whether the test query is similar to the training queries. Concretely, our definitions are:

- Interpolation occurs for a test query when there is at least one similar or duplicate training query.
- Extrapolation occurs for a test query when there is no similar or duplicate training query.

We acknowledge that there can be different implementations of computing query similarity. For example, it can be annotated in terms of intents, entities, etc, or can be computed automatically based on the numbers of matched terms or embedding distance.

In light of the important role of queries in ad-hoc search, they are exclusively considered in our definitions of interpolation and extrapolation, but documents can also be integrated. In modern search engines, not only queries but also documents are dynamically changing. New documents are constantly indexed, and the ranking systems should be capable of generalizing to them which may differ distinctly from the old ones. In our study, we view the corpus as a static set and only consider the dynamics of queries. Further incorporating the dynamics of documents may better characterize the generalization performance when the ranking models are applied in practice. We will explore it in the future.

### 3 INVESTIGATING EXISTING BENCHMARKS

After defining interpolation and extrapolation in IR, now we utilize them to investigate the ad-hoc search benchmarks that are popular in the large data regime of neural IR. MS MARCO dataset [3] and TREC Deep Learning Tracks (TREC DL) [10, 12] are widely-adopted benchmarks for training and evaluating neural ranking models [11].

<sup>2</sup>Some researchers formally define interpolation and extrapolation as whether the test points lie within or outside the convex hull bounded by the extremes of the training data [5, 25, 40, 49]. However, such geometric definition has been challenged by recent studies [4, 54] which show that it is uninformative for the case of high-dimensional spaces and poorly aligns with the generalization performance.

They consist of massive training queries that strongly benefit neural ranking models. Because most neural ranking models since the BERT era [13] are evaluated and compared on the benchmarks, Lin et al. [28] even point out that the rapid progress in IR would not have been possible without them. We will introduce the dataset details and then investigate whether the test-set performance reflects interpolation or extrapolation capacity.

#### 3.1 Benchmark Details

MS MARCO [3] and TREC DL [10, 12] consist of two tightly-related tasks: passage ranking and document ranking, which share very similar training and test queries. MS MARCO transfers passage ranking labels to the document ranking task. TREC DL shares the same test queries between the two tasks. To save the cost of annotation and computation, this paper only focuses on the passage ranking task. But the findings also generalize to the document ranking task due to their tight relation. In the following, we elaborate on the statistics of the passage ranking task.

MS MARCO and TREC DL share the same corpus and training queries. The corpus has approximately 8.8 million passages extracted from Web pages. The training queries are 0.5 million natural-language queries gathered from the Bing search engine's log. They are shallowly annotated. On average, only one passage per query is marked relevant.

MS MARCO and TREC DL provide different test queries. MS MARCO has a development (dev) set and an evaluation set. The latter is hidden and is used for the leaderboard. Thus this paper uses the public dev set for investigation. It has 6,980 natural-language queries sampled from Bing's search log. They are shallowly annotated, and most queries only have one relevant passage. The official evaluation metric is MRR@10. As for TREC DL, it provides small test query sets. The query-passage pairs are judged on a four-point scale. A large annotation pool is constructed to identify a sufficiently comprehensive set of relevant passages. We use the query sets in the year 2019 and 2020. They have 43 and 54 test queries, respectively. The official evaluation metric is NDCG@10.

#### 3.2 Problem of Existing Benchmarks

To investigate whether MS MARCO and TREC DL evaluate the interpolation or extrapolation performance, we examine query similarity between the training and test sets. The examination is carried out in three aspects, i.e., whether two queries share similar entities, similar intents, or the same relevant passages.

Similarity in terms of query entities and intents is examined based on manual annotation. We first use BM25 [39, 53] and an embedding model [55] to separately recall ten training queries for each test query. Then, three annotators are recruited to label the similarity scores between each test query and the recalled training queries. Judgments for entity overlap are on a three-point scale:

- **Full Overlap:** There exists at least one training query that shares exactly the same set of entities with the test query.
- **Partial Overlap:** There exists at least one training query that shares at least one entity with the test query but no training query that fully overlaps with the test query.
- **Zero Overlap:** All recalled training queries do not share any entity with the test query.

**Table 1: Distribution of test queries in terms of entity overlap with training queries. The number shows the percentage of test queries that have exactly/partially/zero (the) same entities with training queries. Dev-samples are 100 queries randomly sampled from MS MARCO dev set.**

| Entity Overlap  | TREC 19 DL | TREC 20 DL | Dev-samples |
|-----------------|------------|------------|-------------|
| Full overlap    | 63%        | 54%        | 45%         |
| Partial overlap | 30%        | 39%        | 51%         |
| Zero overlap    | 7%         | 7%         | 4%          |

**Table 2: Distribution of test queries in terms of intent similarity with training queries. The number shows the percentage of test queries that share duplicate/similar/dissimilar intent with training queries. Dev-samples are 100 queries randomly sampled from MS MARCO dev set.**

| Intent Similarity | TREC 19 DL | TREC 20 DL | Dev-samples |
|-------------------|------------|------------|-------------|
| Duplicate         | 51%        | 37%        | 34%         |
| Similar           | 28%        | 22%        | 19%         |
| Dissimilar        | 21%        | 41%        | 47%         |

Judgments for intent similarity are on a three-point scale:

- **Duplicate:** There exists at least one training query with exactly the same intent as the test query.
- **Similar:** There exists at least one training query with a similar intent as the test query but no duplicate query.
- **Dissimilar:** All recalled training queries are different in intent from the test query.

The average Cohen’s Kappa values [9] are 0.59 and 0.54 for entity and intent annotation, respectively. When there is disagreement among the annotators, we use the median score as the final label.

Annotation results of entity and intent similarity are shown in Table 1 and Table 2, respectively. According to Table 1, we can see that more than 90% test query entities are fully or partially overlapped by one training query. Hence the benchmark is capable of evaluating how ranking models interpolate to previously seen query entities but fails to evaluate how ranking models extrapolate to unseen entities. According to Table 2, we observe a surprisingly high overlap in query intent. More than 50% queries share duplicate or similar intent with training queries. Since we only recall not more than 20 candidates for each test query, the similar training queries for some test queries are missing from the candidates and the true overlap proportions are expected to be higher. Among the three test collections, TREC 19 DL test set is most similar to the training set, where only 21% of test queries are novel. Therefore, all three test collections are biased towards interpolation in terms of intent, and TREC 19 DL is the most biased test set.

We further investigate the training-test query similarity based on overlap in relevant passages. The benchmark corpus consists of short passages that are on average 56 words long. Considering the limited information a short passage contains, two queries that share the same relevant passage are likely to be similar. Moreover, such overlap in relevant passages also leaves the possibility that

**Table 3: Distribution of test queries that share relevant passages with training queries. The numbers show the percentages of test queries that share at least one relevant (somewhat relevant, highly relevant, or perfectly relevant) result with another query in the training set.**

| Passage Relevance             | TREC 19 DL | TREC 20 DL |
|-------------------------------|------------|------------|
| = 3 <b>Perfectly relevant</b> | 26%        | 31%        |
| ≥ 2 <b>Highly relevant</b>    | 60%        | 46%        |
| ≥ 1 <b>Somewhat relevant</b>  | 79%        | 76%        |

**Table 4: Examples of duplicate test and training queries.**

| TREC 20 DL Test Query                                 | Duplicate Training Query                           |
|---|--|
| average wedding dress alteration cost                 | average cost for wedding dress alterations         |
| what does it mean if your tsh is low                  | what does it mean if my tsh is low                 |
| meaning of shebang                                    | what is a shebang                                  |
| do google docs auto save                              | does google docs automatically save                |
| how long does it take to remove wisdom tooth          | how long does it take to have wisdom teeth removed |
| how much would it cost to install my own wind turbine | how much does it cost to install a wind turbine    |
| when did rock n roll begin?                           | when did rock and roll begin                       |

ranking models achieve high test-set performance via memorizing training labels. Now we examine to what extent the test queries of TREC 19 & 20 DL test collections [10, 12] share the same relevant passages with the training queries. Here we focus on TREC DL test sets instead of MS MARCO dev set because it is shallowly annotated and there is little overlap. We mark a test query similar if its any relevant passage is labeled relevant in the training set. Since TREC 19 & 20 DL test collections consist of four-point scaled relevance labels, we convert the labels to binary using different thresholding values. We show the proportion of overlapped test queries in Table 3. It is striking that near 80% queries have at least one somewhat/highly/perfectly relevant passage that is also labeled relevant in the training set. Such a considerable overlap further validates a strong bias towards interpolation evaluation.

Finally, we show several duplicate training-test query pairs. The test queries are manually selected from the TREC 20 DL test set. A subset of duplicate pairs is shown in Table 4. We can see that the training query is very similar to the test query in terms of intents and entities. Most query formulations are alarmingly similar, which only differ in stopwords or minor changes of the word sequence. If a ranking model performs well on these test queries, it may not extrapolate well when applied in the real world where possible queries are numerous. Therefore, it is important to consider interpolation and extrapolation when evaluating neural models that are trained on massive data.

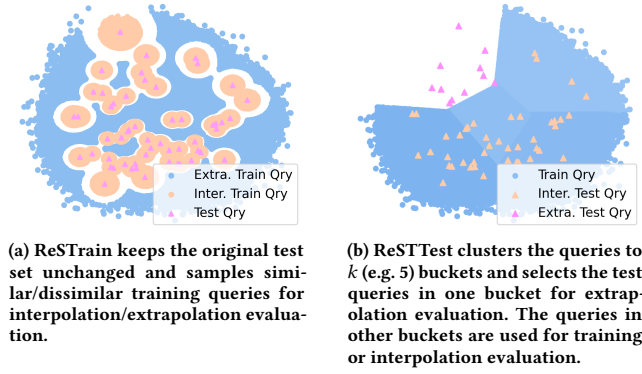


Figure 1: Schematic diagrams of ReSTrain and ReSTTest.

## 4 RESAMPLE-BASED EVALUATION

Having shown that the test queries are too similar to the training ones on benchmarks, we argue that it is important to consider interpolation and extrapolation during evaluation and propose a resampling method to evaluate the model performance in the two regimes. We resample training queries that are similar or dissimilar to the test queries to evaluate interpolation or extrapolation performance, respectively. The proposed resampling method requires neither new test collections nor annotation. It can be directly applied to any Cranfield-like dataset that consists of a training set and a test collection. We first elaborate on the method and then validate its efficacy by investigating whether the results align with the generalization performance evaluated by new out-of-distribution test sets.

### 4.1 Methodology

As introduced in Section 2, we consider the dynamics of queries and define interpolation and extrapolation based on query similarity. Interpolation occurs when the test query is similar to at least one training query. Otherwise, extrapolation occurs. In the following, we will first introduce how we compute query similarity and then utilize the similarity to resample the dataset.

#### 4.1.1 Query Similarity.

There are two options for efficiently computing query similarity, i.e., term-based methods [39] and embedding-based retrieval [55]. We utilize the manual annotation results to decide which is better. Recall that Section 3.2 annotates similarities between test queries and recalled training queries. A better model should recall more similar queries, resulting in higher annotated similarity scores. According to the annotation results, the embedding-based method recalls more similar queries. Hence it is used to compute query similarity when we resample the dataset.

#### 4.1.2 Resampling Training Queries.

The first method is ReSampling Training Queries (ReSTrain). ReSTrain keeps the original test set unchanged. It selects similar or dissimilar training queries as a new training set for interpolation or extrapolation evaluation, respectively. Concretely, ReSTrain collects top- $I$  similar training queries of each test query as the interpolation

training set. It excludes top- $E$  similar training queries of each test query to construct the extrapolation training set. We can tune  $I$  and  $E$  to explore the influence of training set size. We separately train ranking models on the new training sets and evaluate them on the original test set. The gap in the evaluation scores reveals the performance loss when the model transitions from interpolation to extrapolation.

We visualize ReSTrain in Figure 1a. To enable clear visualization, we first use PCA to reduce dimension to two and then resample training queries. Blue and orange points indicate extrapolation and interpolation training queries, respectively. As the figure shows, memorization may help improve interpolation performance because the interpolation training queries are close to the test queries. But it hardly helps extrapolation due to the substantial distinction between training and test queries. Thus, models that do not rely on memorization can achieve similar effectiveness in the two evaluation regimes.

ReSTrain works well for small test sets like TREC 19 & 20 DL because it can find many distinct training queries to construct a new extrapolation training set. Nevertheless, if the test set is large (e.g., MS MARCO dev set), the test queries cover a broad range of topics and thus relate to too many training queries. In this scenario, ReSTrain struggles to evaluate extrapolation performance because only a few truly dissimilar training queries would remain in the extrapolation training set. Next, we will introduce another resampling method to address this limitation.

#### 4.1.3 Resampling Training and Test Queries.

The second evaluation method is ReSampling Training and Test Queries (ReSTTest), which is specially used for large test sets. ReSTTest clusters the large test set to several small ones so that we can evaluate extrapolation performance for a small proportion of the test queries each time.

Specifically, ReSTTest clusters training queries and test queries into  $k$  buckets with  $k$ -means. We use training queries in  $k - 1$  buckets as a new training set. The test queries in those  $k - 1$  buckets are used for interpolation evaluation, and the test queries in the one remaining bucket are used for extrapolation evaluation. We train and evaluate models  $k$  times to acquire full extrapolation results of all test queries. As for interpolation, we report the average evaluation metric score since each test query is judged  $k - 1$  times. In this paper,  $k$  is set to 5.

We visualize ReSTTest in Figure 1b. Like Figure 1a, we also use PCA for dimension reduction to obtain clear visualization. As shown in the figure, we divide the vector space into five ( $k = 5$ ) parts. Each time we select one part. The training queries in the selected part are removed when we train the ranking models. Then, we use the test queries in this part to evaluate the extrapolation capacity and the remaining test queries to evaluate the interpolation performance. We repeat this process five times until all test queries have been used in the extrapolation regime.

## 4.2 Alignment with Generalization

Evaluation of interpolation and extrapolation should align with generalization performance [54]: extrapolation capacity aligns with how well models perform on out-of-distribution test data while interpolation capacity does not. Although we propose an efficient

**Table 5: Performance (NDCG@10) correlation between TREC-Covid and TREC DL. Better correlated results are marked bold except for very low correlation numbers.**

| Datasets    | Domain          | Spearman     |              | Kendall      |              |
|-------------|-----------------|--------------|--------------|--------------|--------------|
|             |                 | Inter.       | Extra.       | Inter.       | Extra.       |
| TREC-Covid  | Bio-Medical     | 0.537        | <b>0.841</b> | 0.359        | <b>0.645</b> |
| NFCorpus    | Bio-Medical     | 0.582        | <b>0.783</b> | 0.418        | <b>0.611</b> |
| NQ          | Wiki-QA         | 0.791        | <b>0.935</b> | 0.606        | <b>0.805</b> |
| HotpotQA    | Wiki-QA         | 0.579        | <b>0.836</b> | 0.416        | <b>0.635</b> |
| DBPedia     | Wiki-Entity     | 0.647        | <b>0.846</b> | 0.492        | <b>0.679</b> |
| CQADupS.    | StackEx. Query  | 0.577        | <b>0.902</b> | 0.403        | <b>0.730</b> |
| Quora       | Quora Query     | 0.655        | <b>0.823</b> | 0.493        | <b>0.647</b> |
| SciDocs     | Sci Citation    | 0.595        | <b>0.724</b> | 0.440        | <b>0.546</b> |
| SciFact     | Sci Citation    | 0.603        | <b>0.793</b> | 0.453        | <b>0.611</b> |
| Arguana     | Counter-argu.   | <b>0.612</b> | 0.504        | <b>0.435</b> | 0.375        |
| FiQA        | Finance-QA      | 0.273        | 0.283        | 0.223        | 0.199        |
| FEVER       | Wiki Fact-check | -0.284       | -0.175       | -0.179       | -0.142       |
| Climate-F.  | Wiki Fact-check | -0.515       | -0.352       | -0.419       | -0.260       |
| Touché-2020 | Argument        | -0.307       | -0.410       | -0.216       | -0.278       |

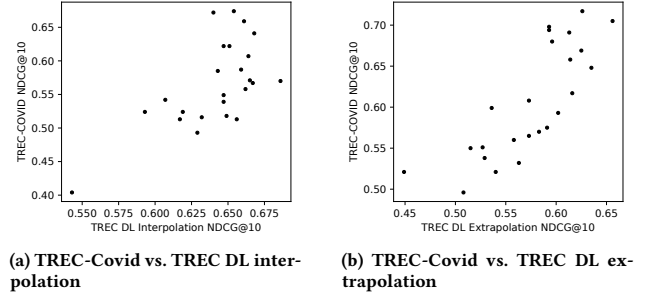
evaluation method that does not require any new testbeds or annotation, the method is limited to a particular distribution that the original dataset follows. It remains to be validated about its alignment with performance on new test sets that follow a different distribution. Therefore, we investigate the alignment between our evaluation results and out-of-distribution generalization performance. We first resample the dataset and train multiple variants of neural retrieval models. We then evaluate their interpolation, extrapolation, and out-of-distribution performance. Finally, we compute the performance correlation to investigate the alignment.

#### 4.2.1 Implementation Details.

To save the computational costs, we opt for ReSTrain in this section. The results should also generalize to ReSTTest because both methods resample queries based on embedding-based similarity. We utilize ReSTrain to resample TREC 19 & 20 DL passage ranking dataset. We resample three sizes of training sets, i.e., 14k, 45k, and 200k training queries. We use BEIR dataset [46] to evaluate the zero-shot performance in unseen domains. Because different neural architectures are affected differently due to the incomplete annotation of BEIR [46], we only use dense retrieval models to alleviate the problem. We follow the common practice to utilize the output vector of [cls] token as text embedding [23, 51, 55]. We use random negative sampling and adopt different pretrained language models for initialization including RoBERTa [31], BERT [13], ERNIE2.0 [45], Condenser [17], coCondenser-wiki [18], and coCondenser-marco [18]. We also initialize the model with BERT [13] and utilize advanced training techniques such as hard negatives [55] and distillation [29]. Overall, there are 8 training methods and 3 training set sizes. We train 24 models on the training sets for interpolation and another 24 models on the training sets for extrapolation. We evaluate the out-of-distribution performance of all 48 models on BEIR.

#### 4.2.2 Discussion.

Performance correlation results are shown in Table 5. We can see that extrapolation performance evaluated on a single benchmark

**Figure 2: Performance (NDCG@10) on TREC-Covid (y-axis) and TREC DL (x-axis).**

dataset strongly correlates with the performance on most out-of-distribution test sets. On the contrary, interpolation performance is substantially less correlated. To present the correlation more clearly, we plot the interpolation and extrapolation correlation in Figures 2a and 2b, respectively. The x-axis shows the performance on the TREC 19 & 20 DL, and the y-axis shows the out-of-distribution performance on TREC-Covid which is regarded as the most reliable test set in BEIR [46, 50]. The two figures are consistent with the correlation results in Table 5. The points in Figure 2b exhibit a strong linear correlation, whereas the points in Figure 2a do not.

We also notice unexpected results in the correlation analysis for some BEIR datasets in Table 5, such as Arguana [48], FiQA [33], FEVER [47], Climate-FEVER [14], and Touché-2020 [7]. Except for FiQA [33] whose paper contains little information about the dataset, we find the rest datasets define relevance differently from typical ad-hoc search. Concretely, Arguana [48] and Touché-2020 [7] are counterargument retrieval and conversational argument retrieval tasks, respectively. FEVER [47] and Climate-FEVER [14] require retrieving evidence that supports given claims. We conjecture that the inconsistent definitions of relevance harm generalization and induce unexpected correlation results.

Overall, the results validate the efficacy of our proposed method. Even if the proposed method is limited to the distribution that the original dataset follows, resampling the dataset based on query similarity still reliably evaluates the extrapolation performance that aligns with the out-of-distribution retrieval effectiveness. Such a result is meaningful because annotating a new test set is hard and costly. Besides, the proposed method additionally evaluates the interpolation capacity, which reflects model behavior in an opposite perspective to extrapolation.

## 5 INTERPOLATION AND EXTRAPOLATION EVALUATION OF EXISTING RETRIEVAL MODELS

Now we investigate the necessity of interpolation and extrapolation evaluation by studying whether evaluation in the two regimes induces different model comparison outcomes. We re-evaluate popular neural retrieval models including various model architectures and training methods.

**Table 6: Interpolation and extrapolation performance on TREC 19 & 20 DL evaluated with ReSTrain.  $\Delta$ =Extra./Inter.-1. #Q denotes the number of training queries. Reranker and ColBERT are interaction-based. SPLADE and dense retrieval (DR) are representation-based.**

| #Q   | Models    | TREC 19 & 20 DL (ReSTrain) |        |          |        |        |          |
|------|-----------|----------------------------|--------|----------|--------|--------|----------|
|      |           | NDCG@10                    |        |          | R@100  |        |          |
|      |           | Inter.                     | Extra. | $\Delta$ | Inter. | Extra. | $\Delta$ |
| 14k  | BM25      | 0.485                      | 0.490  | +1%      | 0.534  | 0.541  | +1%      |
| 14k  | Reranker  | 0.679                      | 0.645  | -5%      | 0.645  | 0.639  | -1%      |
|      | ColBERT   | 0.597                      | 0.585  | -2%      | 0.567  | 0.605  | +7%      |
|      | SPLADE    | 0.596                      | 0.488  | -18%     | 0.588  | 0.547  | -7%      |
|      | DR (base) | 0.617                      | 0.529  | -14%     | 0.576  | 0.534  | -7%      |
| 45k  | Reranker  | 0.692                      | 0.680  | -2%      | 0.659  | 0.650  | -1%      |
|      | ColBERT   | 0.649                      | 0.633  | -2%      | 0.612  | 0.633  | +3%      |
|      | SPLADE    | 0.615                      | 0.512  | -17%     | 0.615  | 0.567  | -8%      |
|      | DR (base) | 0.647                      | 0.558  | -14%     | 0.607  | 0.558  | -8%      |
| 200k | Reranker  | 0.709                      | 0.705  | -1%      | 0.660  | 0.664  | +1%      |
|      | ColBERT   | 0.676                      | 0.661  | -2%      | 0.633  | 0.649  | +3%      |
|      | SPLADE    | 0.637                      | 0.607  | -5%      | 0.642  | 0.611  | -5%      |
|      | DR (base) | 0.664                      | 0.591  | -11%     | 0.621  | 0.564  | -9%      |

## 5.1 Impact of Model Architectures

We compare interaction-based models and representation-based models including dense retrieval and neural sparse retrieval. We first introduce the three model architectures, then brief the implementation details, and finally discuss the evaluation results.

### 5.1.1 Model Architectures.

Interaction-based methods model the term-term interactions to predict relevance. For example, BERT reranker [37] takes the input of both the query and the passage and computes the attention scores between each term pair to contextualize the representations. ColBERT [24] is based on BERT and further leverages a late-interaction mechanism to speed up inference. We use ColBERT [24] and BERT reranker [37] to represent interaction-based models.

Representation-based methods involve dense retrieval and neural sparse retrieval. Dense retrieval takes the input of queries and passages separately and outputs their representations. Neural sparse retrieval predicts term weights and builds inverted indexes for efficient retrieval. It can be regarded as encoding text to a sparse and vocabulary-sized vector. We select the popular SPLADE [16] model as a representative.

### 5.1.2 Implementation.

We use the authors' open-sourced code for training ColBERT [24] and SPLADE [16]. We implement BERT reranker by using pairwise cross-entropy loss. As for dense retrieval, we adopt a base training setting, i.e., 1,024 random negatives per mini-batch and cross-entropy loss. We denote the trained dense retrieval model as DR (base) and will explore other advanced training methods in the next section. All neural models are initialized with BERT-base-uncased [13] and are trained for 50 epochs. For evaluation settings, BERT reranker reranks the top-1000 passages retrieved by BM25, and SPLADE, ColBERT, and dense retrieval end-to-end

**Table 7: Interpolation and extrapolation performance on MS MARCO evaluated with ReSTTest.  $\Delta$ =Extra./Inter.-1. Reranker and ColBERT are interaction-based. SPLADE and dense retrieval (DR) are representation-based.**

| Models    | MSMARCO Dev (ReSTTest) |        |          |        |        |          |
|-----------|------------------------|--------|----------|--------|--------|----------|
|           | MRR@10                 |        |          | R@100  |        |          |
|           | Inter.                 | Extra. | $\Delta$ | Inter. | Extra. | $\Delta$ |
| BM25      | 0.189                  | 0.188  | -1%      | 0.671  | 0.669  | 0%       |
| Reranker  | 0.369                  | 0.346  | -6%      | 0.824  | 0.815  | -1%      |
| ColBERT   | 0.358                  | 0.324  | -9%      | 0.877  | 0.848  | -3%      |
| SPLADE    | 0.324                  | 0.282  | -13%     | 0.867  | 0.812  | -6%      |
| DR (base) | 0.313                  | 0.277  | -12%     | 0.848  | 0.789  | -7%      |

retrieve top passages from the entire corpus. Besides the above neural models, we involve BM25 [53] as a traditional baseline and tune its parameters on the training data with grid search.

We utilize ReSTrain to evaluate interpolation and extrapolation performance on TREC DL test sets. We set the number of training queries to 14k, 45k, and 200k to explore the influence of training data. The results are shown in Table 6. We use ReSTTest to investigate the interpolation and extrapolation performance on the large MS MARCO dev set. All training and test queries are clustered into five buckets. The results are shown in Table 7.

### 5.1.3 Discussion.

According to Tables 6 and 7, we can see that representation-based models, including SPLADE and dense retrieval, suffer from a substantial loss in ranking performance when it transitions from interpolation to extrapolation. Take dense retrieval for example. On TREC DL test sets in Table 6, the NDCG@10 values decrease more than 10%, and R@100 numbers decrease about 8%. The trends are similar on the MSMARCO Dev set in Table 7. On the contrary, interaction-based models are much more robust to extrapolation. BERT reranker and ColBERT extrapolate almost as well as they interpolate in terms of both metrics on TREC DL (Table 6) and R@100 on MSMARCO Dev set (Table 7). Therefore, representation-based models seem to heavily rely on interpolation and are vulnerable to extrapolation. Interaction-based models benefit from modeling term-term interactions and extrapolate well to novel test queries.

Results also suggest that dense retrieval and SPLADE are able to interpolate even similarly to interaction-based models but extrapolate substantially worse. On TREC DL test sets, dense retrieval and SPLADE perform comparably with ColBERT when trained on the interpolation training set. But they substantially underperform when extrapolating. On the MSMARCO Dev set, dense retrieval and SPLADE even outperform the BM25-BERT reranking system in terms of Recall by a large margin when they interpolate. But during extrapolation evaluation, BM25-BERT performs better.

Therefore, interpolation evaluation and extrapolation evaluation may draw different conclusions about which model is better. In a constrained scenario where all possible queries are limited, representation-based models excel because of good interpolation ability and high efficiency. Instead, if the models are required to be applied in search engines where queries are dynamically changing, interaction-based models excel due to strong extrapolation capacity.



**Table 8: Interpolation and extrapolation ranking performance of dense retrieval models (DR) trained with different methods.  $\Delta$ =Extra./Inter.-1. The training set size for ReSTrain is 45k.**

| Type         | PLM         | Finetune | TREC 19 & 20 DL (ReSTrain) |        |          |        |        |          | MS MARCO Dev (ReSTTest) |        |          |        |        |          |
|--------------|-------------|----------|----------------------------|--------|----------|--------|--------|----------|-------------------------|--------|----------|--------|--------|----------|
|              |             |          | NDCG@10                    |        |          | R@100  |        |          | MRR@10                  |        |          | R@100  |        |          |
|              |             |          | Inter.                     | Extra. | $\Delta$ | Inter. | Extra. | $\Delta$ | Inter.                  | Extra. | $\Delta$ | Inter. | Extra. | $\Delta$ |
| DR (base)    | BERT        | RandNeg  | 0.647                      | 0.558  | -14%     | 0.607  | 0.558  | -8%      | 0.313                   | 0.277  | -12%     | 0.848  | 0.789  | -7%      |
| DR (advance) | BERT        | HardNeg  | 0.661                      | 0.581  | -12%     | 0.620  | 0.558  | -10%     | 0.334                   | 0.289  | -13%     | 0.866  | 0.801  | -8%      |
|              | BERT        | Distill  | 0.656                      | 0.573  | -13%     | 0.586  | 0.554  | -5%      | 0.329                   | 0.291  | -12%     | 0.852  | 0.797  | -6%      |
|              | coCon-marco | RandNeg  | 0.668                      | 0.626  | -6%      | 0.641  | 0.644  | 0%       | 0.323                   | 0.299  | -7%      | 0.875  | 0.850  | -3%      |

## 5.2 Impact of Training Strategies

We now re-evaluate various training techniques to explore whether their effectiveness also differs in the two regimes. We consider hard negative mining [51, 55], distillation [21, 29], and pretraining [17, 18]. They are originally proposed to improve effectiveness of dense retrieval models and are also demonstrated to be effective for interaction-based models [43] and neural sparse retrieval [15]. In this paper, we evaluate their efficacy based on dense retrieval models.

### 5.2.1 Advanced Training Methods.

There are mainly three methods to improve the efficacy of dense retrieval. First, Xiong et al. [51] and Zhan et al. [55] employ hard negatives during training. Xiong et al. [51] asynchronously sample hard negatives, and Zhan et al. [55] further propose synchronous sampling via freezing the document index. Second, Hofstätter et al. [21] and Lin et al. [29] propose to distill BERT rerankers and ColBERT, respectively. The two interaction-based models exhibit strong ranking performance and thus are suitable teachers for dense retrieval. Distilling ColBERT is more computationally friendly because ColBERT is more efficient than BERT reranker. Finally, pretraining methods tailored for dense retrieval are also effective [17, 18]. Condenser [17] utilizes an auto-encoding task, and coCondenser [18] leverages contrastive learning.

### 5.2.2 Implementation.

We follow STAR and ADORE [55] to implement hard negative mining. And we follow TCT-ColBERT [29] to implement distillation. The metric scores when we use the full training set match the numbers reported in previous studies, demonstrating the correctness of our implementation. As for pretraining techniques, we explore a variety of pretrained language models (PLM), including BERT [13], RoBERTa [31], ERNIE 2.0 [45], Condenser [17], and coCondenser [18]. All models except coCondenser are trained on Wikipedia corpus. coCondenser has two variants. One denoted as coCon-wiki is pretrained on Wikipedia. The other denoted as coCon-marco is pretrained on Wikipedia and MS MARCO.

### 5.2.3 Comparing Pretraining and Finetuning.

This section investigates how pretraining and finetuning help improve interpolation and extrapolation. The results are shown in Table 8. We can see that initializing models with coCon-marco [17] is the most effective way to improve extrapolation capacity. It substantially reduces the interpolation-extrapolation performance gap, whereas distillation [29] and hard negative mining [55] cannot. It also remarkably improves the absolute extrapolation performance.

**Table 9: Ranking performance of dense retrieval models when initialized by different PLMs.  $\Delta$ =Extra./Inter.-1. The training set size for ReSTrain is 45k. TE and TC stand for text-encoding and target-corpus pretraining, respectively.**

| PLM         | TE | TC | TREC 19&20 DL R@100 (ReSTrain) |        |          | MSMARCO Dev R@100 (ReSTTest) |        |          |
|-------------|----|----|--------------------------------|--------|----------|------------------------------|--------|----------|
|             |    |    | Inter.                         | Extra. | $\Delta$ | Inter.                       | Extra. | $\Delta$ |
| Rand        |    |    | 0.437                          | 0.305  | -30%     | 0.723                        | 0.606  | -16%     |
| RoBERTa     |    |    | 0.582                          | 0.499  | -14%     | 0.845                        | 0.778  | -8%      |
| BERT        | ✓  |    | 0.607                          | 0.558  | -8%      | 0.848                        | 0.789  | -7%      |
| ERNIE2.0    | ✓  |    | 0.610                          | 0.568  | -7%      | 0.856                        | 0.812  | -5%      |
| Condenser   | ✓  |    | 0.615                          | 0.580  | -6%      | 0.859                        | 0.811  | -6%      |
| coCon-wiki  | ✓  |    | 0.615                          | 0.582  | -5%      | 0.862                        | 0.819  | -5%      |
| coCon-marco | ✓  | ✓  | 0.641                          | 0.644  | 0%       | 0.875                        | 0.850  | -3%      |

For example, measured by R@100, its extrapolation performance nearly matches or even outperforms the other models' interpolation performance. On the contrary, hard negative mining and distillation are effective for interpolation. For example, they interpolate better than coCon-marco in MRR@10 on the MS MARCO dev set.

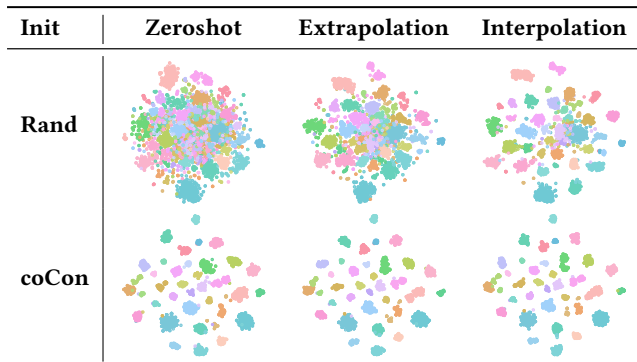
Therefore, the results imply that pretraining is more effective than finetuning in improving the extrapolation ability. It again verifies the importance to evaluate interpolation and extrapolation performance separately. Considering the impressive performance of coCon-marco, we decide to perform a deep dive into how pretraining helps extrapolation.

### 5.2.4 Investigation of Pretraining.

We comprehensively study the efficacy of pretraining to gain insights into what pretraining techniques are effective for improving extrapolation performance. We initialize encoders with various pretrained language models (PLM) and then finetune them using the base training setting, i.e., 1024 random negatives per training step and cross-entropy loss. We also directly finetune a randomly initialized model as a baseline. The results are shown in Table 9.

First, the results highlight the importance of pretrained language models. Random initialization leads to 30% and 16% performance gap on TREC DL and MS MARCO dev set, respectively, which is about twice the gap when we use RoBERTa [31] for initialization. Table 10 visualizes the passage representations of models with and without pretraining. Different colors indicate relevant passages for different TREC 20 DL test queries. We can see that a finetuned random-initialized model (Rand) clusters the passage





**Table 10: Passage representation distribution in different evaluation paradigms of a (finetuned) randomly-initialized model or coCondenser-marco model. Different colors indicate relevant passages for different queries.**

representations poorly and appropriately in the extrapolation and interpolation regimes, respectively. Instead, the pretrained language model (coCon-marco) is already equipped with the representation ability and gets better clustering results than the finetuned random-initialized model. It also substantially reduces the gap between interpolation and extrapolation as the clustering results barely change when moving from interpolation to extrapolation.

Second, it is important to adopt a text-encoding pretraining task. RoBERTa does not use such a task and only pretrains the model on word level. It achieves the worst extrapolation capacity among all pretrained language models. Others adopt different text-encoding tasks, such as next sentence prediction (BERT), sentence ordering prediction (ERNIE2.0), auto-encoding (Condenser), and contrastive learning (coCon-wiki & coCon-marco).

Third, pretraining on the target corpus is beneficial. Both coCon-marco and coCon-wiki adopt contrastive learning for pretraining, while coCon-marco is additionally pretrained on the target corpus and extrapolates substantially better. However, pretraining on the target corpus may be less effective if the definitions of interpolation and extrapolation further consider the dynamics of documents, i.e., some documents are not available during pretraining or finetuning. We will explore it in the future.

Overall, the separation of interpolation and extrapolation helps deeply understand the effectiveness of different methods. Results show that finetuning is beneficial for interpolation and suitable pretraining techniques have a pronounced effect on improving extrapolation capacity.

## 6 RELATED WORK

We recap related work in neural retrieval and studies that investigate its generalization ability. Recent efforts in improving the effectiveness of ranking models can be roughly classified into two directions, i.e., model architectures and training techniques. Popular model architectures involve ColBERT [24, 43], dense retrieval [18, 29, 51, 55] and neural sparse retrieval [15, 16, 27]. ColBERT is interaction-based, while neural sparse retrieval and dense retrieval are representation-based. Effective training techniques include hard

negative mining [51, 55], knowledge distillation [21, 29], and pre-training [17, 32]. They can be combined in practice. Recently, several researchers challenge the generalization ability of neural retrieval models by constructing an out-of-distribution test set. Sciavolino et al. [44] propose a new entity-rich dataset and find dense retrieval performs poorly. Thakur et al. [46] introduce BEIR dataset which contains multiple test sets from different domains. Although proposing a new out-of-distribution test set enables evaluating generalization performance, annotating such a test set is costly. Different from previous studies that evaluate how models generalize on an out-of-distribution test set, we try to take a closer look at the generalization performance without a new test set by analyzing model behavior in the interpolation and extrapolation regimes. This is done by resampling the dataset where models are trained.

## 7 CONCLUSIONS

In this paper, we propose to separately evaluate interpolation and extrapolation capabilities of neural retrieval models. Considering the dynamics of queries in Web search, we define them based on whether the training and test queries are similar or not. Based on the definition, we investigate the bias in popular benchmarks, design associated evaluation methods, and revisit existing neural ranking models. We observe that the popular benchmarks are biased towards interpolation and thus may not reflect how models extrapolate. With the proposed evaluation methods, we find different model architectures and training techniques perform differently in the two regimes. Therefore, it is necessary to consider the two evaluation regimes in future IR studies.

Our work provides a new perspective to evaluate neural retrieval models in the large data regime. It complements previous IR evaluation studies by further considering the training-test relationship. There is plenty to do in this direction. First, since ranking models should be effective for not only unseen queries but also new documents, dynamics of documents can also be incorporated into the definitions of interpolation and extrapolation. Second, this paper defines interpolation and extrapolation without considering the specific ranking models. It would be interesting to explore whether there should be custom definitions for different models to better align with the generalization performance. Third, optimizing ranking models and evaluating extrapolation performance can be regarded as a min-max game. How to design a ranking model that is robust to extrapolation remains to be investigated.

## ACKNOWLEDGMENTS

This work is supported by the Natural Science Foundation of China (Grant No. 61732008) and Tsinghua University Guoqiang Research Institute.

## REFERENCES

- [1] Leif Azzopardi, Paul Thomas, and Nick Craswell. 2018. Measuring the utility of search engine result pages: an information foraging based measure. In *The 41st international acm sigir conference on research & development in information retrieval*. 605–614.
- [2] Peter Bailey, Alistair Moffat, Falk Scholer, and Paul Thomas. 2015. User variability and IR system evaluation. In *Proceedings of The 38th International ACM SIGIR conference on research and development in Information Retrieval*. 625–634.
- [3] Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, et al. 2016. Ms marco: A human generated machine reading comprehension dataset. *arXiv preprint arXiv:1611.09268* (2016).
- [4] Randall Balestriero, Jerome Pesenti, and Yann LeCun. 2021. Learning in high dimension always amounts to extrapolation. *arXiv preprint arXiv:2110.09485* (2021).
- [5] Pietro Barbiero, Giovanni Squillero, and Alberto Tonda. 2020. Modeling generalization in machine learning: A methodological and computational study. *arXiv preprint arXiv:2006.15680* (2020).
- [6] Etienne Barnard and LFA Wessels. 1992. Extrapolation and interpolation in neural network classifiers. *IEEE Control Systems Magazine* 12, 5 (1992), 50–53.
- [7] Alexander Bondarenko, Maik Fröbe, Meriem Beloucif, Lukas Gienapp, Yamen Ajjour, Alexander Panchenko, Christian Biemann, Benno Stein, Henning Wachsmuth, Martin Potthast, and Matthias Hagen. 2020. Overview of Touché 2020: Argument Retrieval. In *CLEF*.
- [8] Cyril Cleverdon and EM Keen. 1966. Aslib–Cranfield research project. *Factors determining the performance of indexing systems* 1 (1966).
- [9] Jacob Cohen. 1960. A Coefficient of Agreement for Nominal Scales. *Educational and Psychological Measurement* 20 (1960), 37–46.
- [10] Nick Craswell, Bhaskar Mitra, Emine Yilmaz, Daniel Campos, and Ellen M. Voorhees. 2020. Overview of the TREC 2019 deep learning track. In *Text REtrieval Conference (TREC)*. TREC.
- [11] Nick Craswell, Bhaskar Mitra, Emine Yilmaz, Daniel Fernando Campos, and Jimmy J. Lin. 2021. MS MARCO: Benchmarking Ranking Models in the Large-Data Regime. *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval* (2021).
- [12] Nick Craswell, Bhaskar Mitra, Emine Yilmaz, Daniel Fernando Campos, and Ellen M. Voorhees. 2020. Overview of the TREC 2020 Deep Learning Track. *ArXiv abs/2102.07662* (2020).
- [13] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 4171–4186.
- [14] Thomas Diggelmann, Jordan Boyd-Graber, Jannis Bulian, Massimiliano Ciaramita, and Markus Leippold. 2020. Climate-fever: A dataset for verification of real-world climate claims. *arXiv preprint arXiv:2012.00614* (2020).
- [15] Thibault Formal, Carlos Lassance, Benjamin Piwowarski, and Stéphane Clinchant. 2021. SPLADE v2: Sparse lexical and expansion model for information retrieval. *arXiv preprint arXiv:2109.10086* (2021).
- [16] Thibault Formal, Benjamin Piwowarski, and Stéphane Clinchant. 2021. SPLADE: Sparse lexical and expansion model for first stage ranking. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2288–2292.
- [17] Luyu Gao and Jamie Callan. 2021. Condenser: a Pre-training Architecture for Dense Retrieval. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. 981–993.
- [18] Luyu Gao and Jamie Callan. 2021. Unsupervised corpus aware language model pre-training for dense passage retrieval. *arXiv preprint arXiv:2108.05540* (2021).
- [19] Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. 2020. Realm: retrieval-augmented language model pre-training. *arXiv preprint arXiv:2002.08909* (2020).
- [20] Pamela J Haley and DONALD Soloway. 1992. Extrapolation limitations of multilayer feedforward neural networks. In *[Proceedings 1992] IJCNN International Joint Conference on Neural Networks*, Vol. 4. IEEE, 25–30.
- [21] Sebastian Hofstätter, Sheng-Chieh Lin, Jheng-Hong Yang, Jimmy Lin, and Allan Hanbury. 2021. Efficiently Teaching an Effective Dense Retriever with Balanced Topic Aware Sampling. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '21)*. 113–122.
- [22] Kalervo Järvelin and Jaana Kekäläinen. 2002. Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems (TOIS)* 20, 4 (2002), 422–446.
- [23] Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense Passage Retrieval for Open-Domain Question Answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.
- [24] O. Khattab and M. Zaharia. 2020. ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT. *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval* (2020).
- [25] David Krueger, Ethan Caballero, Joern-Henrik Jacobsen, Amy Zhang, Jonathan Binas, Dinghui Zhang, Remi Le Priol, and Aaron Courville. 2021. Out-of-distribution generalization via risk extrapolation (rex). In *International Conference on Machine Learning*. PMLR, 5815–5826.
- [26] Brenden M Lake, Tomer D Ullman, Joshua B Tenenbaum, and Samuel J Gershman. 2017. Building machines that learn and think like people. *Behavioral and brain sciences* 40 (2017).
- [27] Jimmy Lin and Xueguang Ma. 2021. A Few Brief Notes on DeepImpact, COIL, and a Conceptual Framework for Information Retrieval Techniques. *arXiv preprint arXiv:2106.14807* (2021).
- [28] Jimmy J. Lin, Rodrigo Nogueira, and Andrew Yates. 2021. Pretrained Transformers for Text Ranking: BERT and Beyond. *Proceedings of the 14th ACM International Conference on Web Search and Data Mining* (2021).
- [29] Sheng-Chieh Lin, Jheng-Hong Yang, and Jimmy Lin. 2020. Distilling Dense Representations for Ranking using Tightly-Coupled Teachers. *arXiv preprint arXiv:2010.11386* (2020).
- [30] Yiding Liu, Weixue Lu, Suqi Cheng, Daiting Shi, Shuaiqiang Wang, Zhicong Cheng, and Dawei Yin. 2021. Pre-trained language model for web-scale retrieval in baidu search. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 3365–3375.
- [31] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: a robustly optimized BERT pretraining approach. *arXiv preprint arXiv:1907.11692* (2019).
- [32] Shuqi Lu, Chenyan Xiong, Di He, Guolin Ke, Waleed Malik, Zhicheng Dou, Paul Bennett, Tieyan Liu, and Arnold 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*. 2780–2791.
- [33] Macedo Maia, Siegfried Handschuh, André Freitas, Brian Davis, Ross McDermott, Manel Zarrouk, and Alexandra Balahur. 2018. WWW'18 Open Challenge: Financial Opinion Mining and Question Answering. *Companion Proceedings of the The Web Conference 2018* (2018).
- [34] Antonio Mallia, Omar Khattab, Torsten Suel, and Nicola Tonello. 2021. Learning Passage Impacts for Inverted Indexes. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '21)*.
- [35] Jiaxin Mao, Tetsuya Sakai, Cheng Luo, Peng Xiao, Yiqun Liu, and Zhicheng Dou. 2019. Overview of the NTCIR-14 we want web task. *Proceedings of NTCIR-14* (2019), 455–467.
- [36] Alistair Moffat and Justin Zobel. 2008. Rank-biased precision for measurement of retrieval effectiveness. *ACM Transactions on Information Systems (TOIS)* 27, 1 (2008), 1–27.
- [37] Rodrigo Nogueira and Kyunghyun Cho. 2019. Passage Re-ranking with BERT. *arXiv preprint arXiv:1901.04085* (2019).
- [38] Yingqi Qu, Yuchen Ding, Jing Liu, Kai Liu, Ruiyang Ren, Wayne Xin Zhao, Daxiang Dong, Hua Wu, and Haifeng 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*. 5835–5847.
- [39] Stephen E. Robertson and Karen Spärck Jones. 1976. Relevance weighting of search terms. *JASIS* 27, 3 (1976), 129–146. <https://doi.org/10.1002/asi.4630270302>
- [40] Elan Rosenfeld, Pradeep Ravikumar, and Andrej Risteski. 2022. An online learning approach to interpolation and extrapolation in domain generalization. In *International Conference on Artificial Intelligence and Statistics*. PMLR, 2641–2657.
- [41] Shiori Sagawa, Pang Wei Koh, Tatsunori B Hashimoto, and Percy Liang. 2019. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. *arXiv preprint arXiv:1911.08731* (2019).
- [42] Tetsuya Sakai, Sijie Tao, Zhaohao Zeng, Yukun Zheng, Jiaxin Mao, Zhumin Chu, Yiqun Liu, Maria Maistro, Zhicheng Dou, Nicola Ferro, et al. 2020. Overview of the NTCIR-15 we want web with CENTRE (WWW-3) task. *Proceedings of NTCIR-15. to appear* (2020).
- [43] Keshav Santhanam, Omar Khattab, Jon Saad-Falcon, Christopher Potts, and Matei Zaharia. 2021. ColBERTv2: Effective and Efficient Retrieval via Lightweight Late Interaction. *arXiv preprint arXiv:2112.01488* (2021).
- [44] Christopher Scialovino, Zexuan Zhong, Jinhyuk Lee, and Danqi Chen. 2021. Simple Entity-Centric Questions Challenge Dense Retrievers. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. 6138–6148.
- [45] Yu Sun, Shuohuan Wang, Yukun Li, Shikun Feng, Hao Tian, Hua Wu, and Haifeng Wang. 2020. ERNIE 2.0: A Continual Pre-training Framework for Language Understanding. *ArXiv abs/1907.12412* (2020).
- [46] Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna 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)*.

- [47] James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. Fever: a large-scale dataset for fact extraction and verification. *arXiv preprint arXiv:1803.05355* (2018).
- [48] Henning Wachsmuth, Shahbaz Syed, and Benno Stein. 2018. Retrieval of the best counterargument without prior topic knowledge. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 241–251.
- [49] Taylor Webb, Zachary Dulberg, Steven Frankland, Alexander Petrov, Randall O'Reilly, and Jonathan Cohen. 2020. Learning representations that support extrapolation. In *International conference on machine learning*. PMLR, 10136–10146.
- [50] Ji Xin, Chenyan Xiong, Ashwin Srinivasan, Ankita Sharma, Damien Jose, and Paul N Bennett. 2021. Zero-Shot Dense Retrieval with Momentum Adversarial Domain Invariant Representations. *arXiv preprint arXiv:2110.07581* (2021).
- [51] Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul N. Bennett, Junaid Ahmed, and Arnold Overwijk. 2021. Approximate Nearest Neighbor Negative Contrastive Learning for Dense Text Retrieval. In *International Conference on Learning Representations*. <https://openreview.net/forum?id=zeFrfgYzIn>
- [52] Keyulu Xu, Mozhi Zhang, Jingling Li, Simon S Du, Ken-ichi Kawarabayashi, and Stefanie Jegelka. 2020. How neural networks extrapolate: From feedforward to graph neural networks. *arXiv preprint arXiv:2009.11848* (2020).
- [53] Peilin Yang, Hui Fang, and Jimmy Lin. 2018. Anserini: Reproducible ranking baselines using Lucene. *Journal of Data and Information Quality (JDIQ)* 10, 4 (2018), 1–20.
- [54] Claudio Zeni, Andrea Anelli, Aldo Glielmo, and Kevin Rossi. 2022. Exploring the robust extrapolation of high-dimensional machine learning potentials. *Physical Review B* 105, 16 (2022), 165141.
- [55] Jingtao Zhan, Jiaxin Mao, Yiqun Liu, Jiafeng Guo, Min Zhang, and Shaoping Ma. 2021. Optimizing Dense Retrieval Model Training with Hard Negatives. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '21)*. 1503–1512.
- [56] Jingtao Zhan, Jiaxin Mao, Yiqun Liu, Jiafeng Guo, Min Zhang, and Shaoping 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*. 1328–1336.
- [57] Fan Zhang, Yiqun Liu, Xin Li, Min Zhang, Yinghui Xu, and Shaoping Ma. 2017. Evaluating web search with a bejeweled player model. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 425–434.