Fostering Coopetition While Plugging Leaks: The Design and Implementation of the MS MARCO Leaderboards

Jimmy Lin
jimmylin@uwaterloo.ca
University of Waterloo
Canada

Daniel Campos
dcampos3@illinois.edu
University of Illinois
Urbana-Champaign
USA

Nick Craswell
nickcr@microsoft.com
Microsoft
USA

Bhaskar Mitra
bmitra@microsoft.com
Microsoft
Canada

Emine Yilmaz
emine.yilmaz@ucl.ac.uk
University College London
UK

ABSTRACT

We articulate the design and implementation of the MS MARCO document ranking and passage ranking leaderboards. In contrast to "standard" community-wide evaluations such as those at TREC, which can be characterized as simultaneous games, leaderboards represent sequential games, where *every* player move is immediately visible to the entire community. The fundamental challenge with this setup is that *every* leaderboard submission leaks information about the held-out evaluation set, which conflicts with the fundamental tenant in machine learning about separation of training and test data. These "leaks", accumulated over long periods of time, threaten the validity of the insights that can be derived from the leaderboards. In this paper, we share our experiences grappling with this issue over the past few years and how our considerations are operationalized into a coherent submission policy. Our work provides a useful guide to help the community understand the design choices made in the popular MS MARCO leaderboards and offers lessons for designers of future leaderboards.

CCS CONCEPTS

• **Information systems** → **Retrieval effectiveness**.

KEYWORDS

community-wide evaluations, datasets, neural models

ACM Reference Format:
Jimmy Lin, Daniel Campos, Nick Craswell, Bhaskar Mitra, and Emine Yilmaz. 2022. Fostering Coopetition While Plugging Leaks: The Design and Implementation of the MS MARCO Leaderboards. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '22), July 11–15, 2022, Madrid, Spain.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3477495.3531725

SIGIR '22, July 11–15, 2022, Madrid, Spain

ACM ISBN 978-1-4503-8732-3/22/07…$15.00
https://doi.org/10.1145/3477495.3531725

1 INTRODUCTION

At its core an empirical discipline, progress in system-oriented information retrieval research has been driven by evaluations since time immemorial. As William Thomson (otherwise known as Lord Kelvin) famously quipped, "If you cannot measure it, you cannot improve it."[1] This is a lesson that the community has taken to heart since the founding of the field more than half a century ago. Large-scale evaluations have been a fixture in IR for at least three decades: the 30th edition of the Text Retrieval Conferences organized by the U.S. National Institute for Standards and Technology occurred in 2021, and similar efforts around the world such as CLEF, NTCIR, and FIRE are still going strong.

In recent years, large-scale community evaluations have been supplemented by so-called "leaderboards", most prominently in the computer vision and natural language processing communities. The growing popularity of leaderboards tracks with the advent of neural networks in a mutually reinforcing manner: leaderboards are typically accompanied by large datasets, which feed data-hungry neural models that are able to perform well on the defined tasks. High levels of effectiveness attract the attention of researchers, who flock to the task (and data) en masse. Increased participation raises the prestige associated with "owning" the top leaderboard position, which comes with bragging rights of being "state of the art", and around and around we go. While there are undeniable downsides of leaderboard-driven research [4, 16], we believe that they have genuinely led to *real* advances in technology.

In Craswell et al. [7], we shared some perspectives about benchmarking ranking models in the large-data regime with the MS MARCO datasets and the associated efforts around them, including the MS MARCO leaderboards and the TREC Deep Learning Tracks [8, 9]. We discussed issues surrounding internal and external validity, and demonstrated the stability of leaderboard results via bootstrapping experiments. Missing from that paper, however, was a more in-depth discussion of the design and implementation of the leaderboards themselves. This paper attempts to fill that gap.

Leaderboards can be characterized as sequential games, where player moves are immediately visible to the community. This property is a great source of strength, promoting rapid progress in the field, but is simultaneously a big weakness, as each submission leaks

[1] Lord Kelvin wasn't known for being succinct, and it was likely that he never uttered these words—rather, the quote is more of a paraphrase of a much longer statement.

Dataset	$\lvert q\rvert$	$\overline{L}(q)$	$\lvert J\rvert$	$\lvert J\rvert/q$	$\lvert\text{Rel}\rvert/q$
MS MARCO passage ranking (training set)	502,939	6.06	532,761	1.06	1.06
MS MARCO passage ranking (development set)	6,980	5.92	7,437	1.07	1.07
MS MARCO passage ranking (evaluation set)	6,837	5.85	-	-	-
MS MARCO document ranking (training set)	367,013	5.95	367,013	1.0	1.0
MS MARCO document ranking (development set)	5,193	5.89	5,193	1.0	1.0
MS MARCO document ranking (evaluation set)	5,793	5.85	-	-	-

Table 1: Summary statistics of queries and relevance judgments for the MS MARCO datasets: number of queries $|q|$, mean query length $\overline{L}(C)$, number of judgments $|J|$, judgments per query $|J|/q$, and relevant documents per query $|\textbf{Rel}|/q$.

Corpus	$\lvert C\rvert$	$\overline{L}(C)$	$\widetilde{L}(C)$
MS MARCO passage corpus	8,841,823	56.3	50
MS MARCO document corpus	3,213,835	1131.3	584

Table 2: Summary statistics for the MS MARCO passage corpus and document corpus: number of documents $|C|$, mean document length $\overline{L}(C)$, and median document length $\widetilde{L}(C)$.

information about the held-out evaluation set. Balancing these opportunities and challenges has been the primary focus of our efforts. We describe the implementation of the leaderboards themselves, which are two independent instances of a Git-based framework that accepts submissions via pull requests, and how our design considerations are operationalized into concrete policy. Finally, we reflect on a number of lessons learned along the way in running the leaderboards over the past few years.

The contribution of this work is a clear articulation of what we, as organizers, hope to accomplish with the MS MARCO leaderboards, and how our goals are implemented. This helps the community understand our design choices and the rationale behind many policy decisions, which have been the subject of vigorous internal debates—and to date have mostly been buried in private email threads and exist only as tacit knowledge in the minds of the organizers. These discussions also offer more transparency into the issues that are of greatest concern to us. More broadly, we believe that many of the issues we have grappled with are common across other leaderboards and evaluations, and thus we believe that our experiences can provide valuable lessons for other organizers of future leaderboards.

2 DESIGN GOALS

While references to MS MARCO today are strongly associated with the passage ranking dataset (released October 2018) and the document ranking dataset (released May 2019), there are actually a series of "MS MARCO datasets" dating back to 2016 [1, 14], focused on problems that range from question answering to keyphrase extraction. These datasets, however, all share the common goal of helping academic researchers explore information access in the large-data regime [7–9] by making available large amounts of (supervised) training data to feed data-hungry neural models.

The passage ranking and document ranking datasets both capture standard *ad hoc* retrieval tasks, but on different corpora: The passage ranking dataset is built on a corpus of 8.8M paragraph-length texts that are typical of "answers" shown at the top of a search engine's result page. The dataset contains anonymized natural language questions from Bing query logs that explicitly express an informational intent [3], and each query is associated with, on average, one relevant passage (as assessed by human annotators). While the original Question Answering dataset features many unanswerable questions, the *ad hoc* ranking task ignores them and as a result the passage ranking task contains only answerable questions that have at least one relevant passage in the corpus. The queries in the passage dataset are divided into a training set, a development set, and an evaluation set.

The corpus used in the passage ranking task comprises 8.8M short passages extracted from 3.5M URLs by the Bing question answering ranking service between June 2016 and March 2018. The corpus for the document ranking dataset comprises 3.2M full-length web pages that the passages came from. The document corpus was extracted on May 2018 and as a result around 300K URLs were not discoverable. The document ranking dataset likewise had queries divided into training, development, and evaluation sets, but relevance judgments were "transferred" from the passage dataset under the assumption that a document with a relevant passage is a relevant document (although this process was a bit noisy due to the temporal misalignment of the two corpora). Detailed statistics of the queries and relevance judgments for both datasets are shown in Table 1, and detailed statistics of the corpora are shown in Table 2. In these analyses, token counts are computed by splitting texts on whitespace using Python's `split()` method for strings.

There are two independent MS MARCO leaderboards, one for passage ranking and another for document ranking. Strictly speaking, the leaderboard results refer to scores on the held-out evaluation queries, although participants are asked to submit results for the development set as well. The evaluation queries are publicly available, but the relevance judgments are withheld from the public and scores can only be obtained via a leaderboard submission. The passage leaderboard "opened" at around the same time the passage ranking dataset was released, in October 2018. However, the document ranking leaderboard did not "open" until August 2020; prior to that time, the corpus as well as the training and development sets were available, but not the evaluation queries.

The MS MARCO datasets were also used in the Deep Learning Tracks at TREC 2019 [9] and TREC 2020 [8], with the goal of comparing evaluation results based on so-called "sparse" judgments

from the leaderboard (i.e., many queries but few relevance judgments per query, characteristic of the MS MARCO training data) with "dense" judgments that are gathered via pooling, as is typical in TREC evaluations. While these comparisons have been quite insightful and have contributed to IR evaluation methodology, the MS MARCO leaderboards and the TREC Deep Learning Tracks are distinct evaluations.

The goal of the MS MARCO leaderboards is to encourage "coopetition", a portmanteau of cooperation and competition, among various academic and industry research groups working on deep learning and other methods that require or benefit from large-scale (supervised) training data. While we encourage friendly competition between different participating groups for top positions on the leaderboard, our core motivation is to ensure that over time the leaderboard provides meaningful scientific insights about how different methods compare to each other and answer questions like whether we are making real progress as a research community. We hope that all participants abide by this spirit of coopetition and strictly observe good scientific principles when participating. We follow an honor system and expect that participants comply with the spirit of this approach and our submission policy.

2.1 How Are Leaderboards Different?

We begin with the obvious question: How are leaderboards different from "standard" community-wide evaluations such as those at TREC that have been going on for decades? In our specific case, there is a synergistic relationship between the MS MARCO leaderboards and the TREC Deep Learning Tracks as they together provide methodological insights into large-scale evaluations, as discussed above. However, let's set aside these methodological explorations for this discussion.

At a high level, leaderboards and "standard" community-wide evaluations share many common features: in both cases, there are common datasets and evaluation resources. Both attract participation from researchers and practitioners. System outputs are gathered by organizers and evaluated according to some pre-defined guidelines. Results are disseminated, and there is invariably some listing of the effectiveness of the participants' submissions, sorted by a figure of merit. Participants may decide to share their technical approach with the community, in forms of documentation ranging from blog posts to research papers.

There is, however, one key distinction: Whereas community-wide evaluations are temporally well-defined with a fixed submission date, participation in leaderboards is continuous, with submissions asynchronously arriving over a much longer period of time (can be years). In other words, leaderboards are sequential games, as opposed to "standard" evaluations, which are simultaneous games. This key difference is a great source of strength but a critical weakness as well. The sequential nature of leaderboard submissions gives rise to a number of important design considerations that set leaderboards apart from other types of evaluations. We turn our attention to these issues next.

2.2 Implications for Design

The ability to submit results to a leaderboard at any time is more conducive to rapid progress than the typical annual cycle of most community-wide evaluations. This is especially true in the modern era of deep learning, where advances occur in the timeframe of months, and a model might become obsolete in six months. In this context, leaderboards provide a vehicle by which progress can be continuously evaluated. Waiting for the next yearly TREC cycle, for example, would be unappealing for many researchers, and this slower pace may retard technical progress.

However, there is a tremendous danger—and it is addressing this single challenge that has perhaps occupied most of our efforts as organizers. Separation of training from test data is one of the most basic tenants of machine learning and any solution based on machine learning techniques. The challenge with leaderboards is that, as a sequential game, *every* player move (i.e., submission by a participant) is immediately visible to all players and leaks information about the held-out evaluation set. Since *every* submission is assessed on the held-out evaluation set in short order, "what works" and "what doesn't" immediately becomes public knowledge—to the extent that the submissions are descriptive with respect to the technique employed. These "leaks", accumulated over time, threaten the validity of the scientific insights that can be meaningfully derived from the leaderboard. In this way, leaderboards can become victims of their own success: as a leaderboard gains popularity, the cumulative effect of these leaks become greater as well.

In contrast, community-wide evaluations that operate within a well-defined submission window have far more stringent methodological safeguards. As a simultaneous game, all participants get "one shot" to make their play, oblivious to the submissions of every other participant. Moreover, each team is only allotted a small number of submissions (typically, three), which eliminates the possibility of (inadvertently) tuning on test data—for example, the methodologically poor practice of evaluating a large number of models and only selectively reporting results of those that performed well. Furthermore, the literature clearly distinguishes between submissions that were officially submitted to the evaluation and results that were obtained after evaluation resources became available (so-called post-hoc runs). Generally, the latter runs are afforded "lower status" in terms of the veracity of their results.

In organizing the MS MARCO leaderboards over the past few years, we have observed a number of behavioral patterns that we feel are not consonant with the spirit of coopetition, discussed below in detail. These can be viewed as lessons that we learned over time, as many of these issues were not anticipated at the beginning. To be clear, we are *not* accusing any participant of willful untoward behavior. We can only assume that all participants are operating in good faith and share our desire to "do good science". However, we have observed some submissions and some participant behavior that have given us pause, which means that our concerns *are* founded in reality. Nevertheless, for clarity of presentation and to illustrate specific points, the descriptions below may be somewhat exaggerated.

Tuning on the held-out evaluation set. The sequential nature of leaderboard submissions means that it possible to (inadvertently) tune on held-out data. Consider a hypothetical set of submissions that differed only in the random seed used to initialize the training of a neural model. Naturally, we would expect some variation in the effectiveness of these model variants [6]. If the participant then

reported only the highest score (e.g., in a research paper), no doubt there would be agreement that this represents tuning on the test set, thus making the results highly questionable.

Similar dangers exist when a participant submits multiple models in a sequential process of experimenting with different variants. If such variants are submitted one after the other, it is unavoidable that later submissions will be influenced by the effectiveness of previous results. The participant would be, in effect, performing model selection and hill-climbing on the evaluation set. This would be methodologically similar to conducting a line search in the design space based on scores on held-out test data.

An even more subtle form of this behavior can inadvertently occur across different groups, and is unfortunately enabled by another desirable properly that we value as a community: reproducibility. Consider the case when a group atop the leaderboard shares code so that the top-scoring run can be easily reproduced. This means that anyone else can build on those results, perhaps making minor modifications, and submit their own runs. On the plus side, these runs increase the veracity of the proposed method since multiple teams "got it to work". On the down side, it may be difficult to tell whether subsequent minor modifications just "got lucky" in generating improvements, or have made a genuine contribution. Why is this the case? In the simplest scenario, let's suppose that the second group simply reran the existing code, but with a trivial modification such as a different random seed. There is some likelihood that this run outperforms the previous run, perhaps achieving a new state of the art—but needless to say this is just a matter of statistical noise, and cannot be considered a technical advance. Multiple groups behaving this way would be in effect (inadvertently) colluding to tune on the evaluation data. Given the challenges associated with significance testing on leaderboards (see, for example, Lin et al. [12]), how can we separate genuine scientific advances from the general variability associated with neural models?

In the more general case, any frequently used common code base might cause similar issues, because it becomes quite easy to start from a single model and accumulate improvements by observing which techniques succeeded and which failed in other submissions. In a common code base, it is easier to accumulate small improvements based on knowledge of the evaluation set that should not have been accessible. Once again, this is hill-climbing based on held-out data. Thus, reproducibility is potentially an enemy of scientific validity and solid methodological progress in the context of a popular leaderboard.

These considerations led to a concrete policy decision to restrict the frequency of leaderboard submissions, which we feel at least partially alleviates these issues. Furthermore, we explicitly ask participants to refrain from submitting multiple model variants that only differ in minor ways (e.g., in hyperparameters). We detail these policy decisions in Section 3.3.

Proliferation of "uninteresting" techniques. To preserve the spirit of the leaderboard and to ensure the scientific integrity of the results, we wished to actively discourage techniques that we felt were "uninteresting". One hypothetical scenario is a participant who did nothing other than to run a simple ensemble of the top-k leaderboard submissions. Such a run is likely to perpetually sit atop the leaderboard; when a new highly effective run was submitted, the fusion could simply be repeated with the new contribution to further boost effectiveness. To us, such a technique would represent an example of tuning on the held-out evaluation set (since the evaluations scores for individual models would be used for model selection) and would be scientifically uninteresting.

Such a scenario is not idle speculation. In fact, this is exactly what happened towards the end of the Netflix prize, when several of the top performing teams began joining forces to build huge (and complicated) ensembles just to push the scores a bit higher. At that point, we think many would agree that the prize had lost its value in driving *scientific* advances. Furthermore, at that point the methods had also become impractical from an engineering perspective. According to the organizers:[2]

> [the final Grand Prize ensemble] is a truly impressive compilation and culmination of years of work, blending hundreds of predictive models to finally cross the finish line. We evaluated some of the new methods offline but the additional accuracy gains that we measured did not seem to justify the engineering effort needed to bring them into a production environment.

Applying these lessons to our context, we discussed at length whether participant runs should be publicly accessible. This, in fact, is the norm for most community-wide evaluations: NIST, for example, maintains a complete and open archive of all submissions to all tracks at every TREC, made available to the community after the conclusion of each evaluation cycle. These archives provide a valuable resource for meta-evaluations and other studies focused on the evaluation methodology itself. Many such studies have been conducted by researchers outside NIST, and thus this data availability broadens the lines of inquiry that can be pursued by researchers (compared to the alternative scenario where NIST does not make the runs available).

However, there is a key difference here—any insights gained, for example, by analyzing old TREC-8 runs, cannot be applied to generate another TREC-8 run that is directly comparable to the runs that TREC-8 participants submitted. It is obvious that any subsequent result is a post-hoc run, and researchers understand the associated implications (i.e., fewer methodological safeguards). Not so with leaderboards. A public archive of all previous submissions could invite researchers to mine relevance signals from those runs, and the extracted insights could then be exploited to generate better results on the leaderboard. Any such effort would occur with the knowledge of evaluation scores on the held-out test data, once again exploiting "leaked" information. There would be no principled and enforceable way to ensure that this doesn't happen.

Thus, we saw having a public archive of submissions as a threat to the scientific validity of the leaderboard. This led to our decision—contrary to TREC norms—to *not* make historical submissions (the run files) publicly accessible. This is operationalized in encrypted submissions, described in Section 3.

Anonymous probing of evaluation scores. The popularity of the MS MARCO leaderboards has conferred prestige to highly ranked runs. The entry atop the leaderboard earns "bragging rights"

[2] https://netflixtechblog.com/netflix-recommendations-beyond-the-5-stars-part-1-55838468f429

as representing the state of the art, a fact that is frequently touted and discussed in blogs and on social media. Extracts from the leaderboards also appear frequently in research papers to contextualize the effectiveness of a particular model.

Conversely, a poorly performing submission may be viewed as negatively affecting the reputation of the submitting team and its parent organization. As a result of these realities, we have observed organizations that attempt to "probe" the leaderboard and obtain scores on the evaluation set before fully disclosing their identities. This often proceeds as follows: a submission is associated with a nondescript organization (for example, an ambiguous acronym), sometimes from a "burner" account. If the submission performs well, it is followed up by a request to change the organization name (and other metadata) associated with the run, or we receive another run from the same participant with a clear identity.

This particular lesson was not apparent to us until the leaderboard was well under way, and we did not implement a coherent policy solution until (roughly) the fourth quarter of 2020. Prior to this development, there was no clarity regarding anonymous submissions vs. embargoed submissions. In the first case, there is no intention for the participants to reveal their identities. In the second case, the anonymization is only temporary, to facilitate blind review. The current policy is that anonymous submissions are not allowed. The leaderboard allows embargoed submissions—we still gather the identities of the participant (and all relevant metadata), but do not reveal this information until some time after the submission. Furthermore, we provide concrete guidelines on how participants should identify themselves. How these restrictions are operationalized is discussed in Section 3.

3 IMPLEMENTATION

The MS MARCO leaderboards are implemented as self-contained GitHub repositories, where submissions are accepted via pull requests. There are two separate leaderboards, one for passage ranking[3] and the other for document ranking,[4] but both are nearly identical in terms of organization, policy, and procedures.

The high-level idea is that a leaderboard submission follows standard best practices for software development. A participant forks the main leaderboard repository, commits the submission (appropriately packaged, details below) in their local clone, and then sends a pull request to the main trunk. In processing the pull request (details below), the organizers evaluate the submission and then merge the contribution into the main trunk. Thus, the repository also serves as an archive of all historical submissions, which facilitates meta-evaluations and other analyses.

The leaderboard itself is a webpage that is dynamically generated from an underlying CSV file that holds run metadata; see Figure 1 for a screenshot from the document leaderboard. A leaderboard entry shows the date of submission, a description of the run, a description of the team, links to code and paper (if available), the run classification ("full ranking" vs. "reranking"), and the official evaluation scores on the development and evaluation sets. Leaderboard entries are sorted by the official metric on the held-out evaluation set: MRR@10 for the passage ranking task and MRR@100 for the document ranking task. Metrics are rounded to three places, and ties are broken by chronological order; that is, the earlier submission is ranked higher. Finally, a small trophy icon denotes leaderboard entries that were, at the time of submission, the best scoring run on the evaluation set. The sequence of these trophy icons traces advances in the "state of the art" (SOTA) over time.

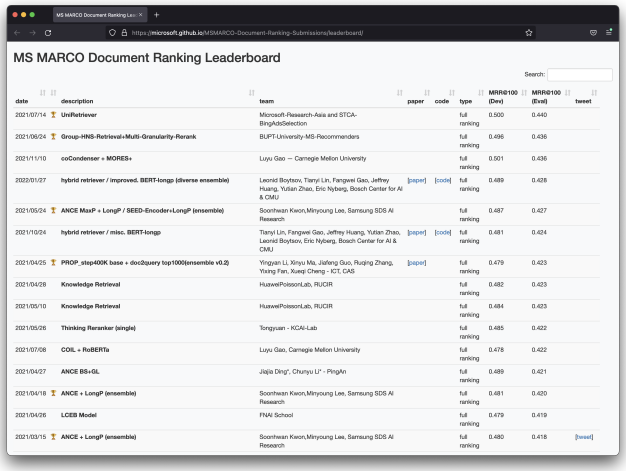


Figure 1: Screenshot of the document ranking leaderboard.

In the remainder of this section, we detail the mechanics of how participants submit runs, how those runs are then processed by the organizers, and our submission policy.

3.1 Submission Mechanics

A submission is made to either the passage or document ranking leaderboard observing the following procedure:

(1) The participant first chooses a submission id, which will be a permanent (public) unique key, akin to the "runtag" in a TREC submission. We ask that the ids take the form `yyyymmdd-foo`, where the prefix is the submission (year, month, day) and `foo` can be an arbitrary alphanumeric string, e.g., representing the participating team's name. No maximum length of the id is enforced, except that we ask participants to keep the length "reasonable".

(2) The participant clones the GitHub repository corresponding to the leaderboard (either passage ranking or document ranking). Note that the individual performing the submission must have a GitHub account, but we allow submissions from anonymous "burner" GitHub accounts (e.g., an account created for the sole purpose of making a submission).

In the directory `submissions/` of the participant's fork, we ask for the following files (illustrative example with the document leaderboard; the process is similar for the passage leaderboard):

- `submissions/yyyymmdd-foo/dev.txt.bz2`: run file on the development queries compressed with bzip2.
- `submissions/yyyymmdd-foo/eval.txt.bz2`: run file on the held-out evaluation queries compressed with bzip2.
- `submissions/yyyymmdd-foo-metadata.json`: the run metadata.

[3] https://github.com/microsoft/MSMARCO-Passage-Ranking-Submissions
[4] https://github.com/microsoft/MSMARCO-Document-Ranking-Submissions

The run files are in a simplified version of the standard TREC format. For passage ranking runs, we ask that participants submit 1000 hits per query; for document ranking runs, we ask that participants submit 100 hits per query. There is no principled reason for this difference; the settings are purely historical idiosyncrasies.

Relevance judgments are publicly available for the development queries, and thus the participant can already evaluate the development run, but we ask for its inclusion in the official submission anyway, for sanity checking and to support downstream analyses. Relevance judgments are not publicly available for the evaluation queries; the effectiveness of this run (MRR@10 for passage ranking and MRR@100 for document ranking) determines the position of the submission on the leaderboard.

The metadata file takes the following format:

```
{
  "team": "team name",
  "model_description": "model description",
  "paper": "url",       // URL to paper
  "code": "url",        // URL to code
 "type": "full ranking" // either 'full ranking' or 'reranking'
}
```

Detailed discussion around participant identities is saved for Section 3.3. We provide no concrete guidance for the model description, and submissions range from descriptive model names to the whimsical. Participants are asked to supply URLs to the corresponding paper and code if available, and indeed for some submissions these fields are non-empty. The "code" field typically points to a GitHub repository, and in some cases, to notebooks that demonstrate the code used to generate the run.

These metadata fields correspond to what is displayed on the leaderboard. To facilitate blind review, we allow embargoed submissions; this is further described in Section 3.3.

(3) We ask that the participant execute our evaluation script to make sure everything is in order (and fix any errors):

```
$ python eval/run_eval.py --id yyyymmdd-foo
```

This is exactly the evaluation script we run (see Section 3.2), with the exception that we have access to the relevance judgments for the held-out evaluation set.

(4) In the penultimate step, the participant packages the submission using the following script:

```
$ eval/pack.sh yyyymmdd-foo
```

This script encrypts the files from step (2) using the MS MARCO leaderboard public key (separate keys for the passage ranking and document ranking leaderboards). Our rationale for *not* making runs publicly accessible has already been discussed in Section 2.2.

In a bit more detail, we adopt the standard practice for sharing encrypted files that first generates a random symmetric key to encrypt the runs and the metadata, and then encrypts the symmetric key itself using our public key.

(5) Finally, the actual submission to the leaderboard is a pull request that comprises the following three files:

- `submissions/yyyymmdd-foo.tar.enc`: the runs that have been packed into a tarball and then encrypted with the symmetric key.
- `submissions/yyyymmdd-foo-metadata.json.enc`: the metadata encrypted with the symmetric key.
- `submissions/yyyymmdd-foo.key.bin.enc`: the symmetric key that has been encrypted with our public key.

Note that the pull request comprises only ciphertext, and that the runs (and metadata) are only readable by the organizers, since we are the only ones in possession of the private key.

3.2 Evaluation Mechanics

The creation of the pull request comprising a submission triggers emails that are sent to the organizers, via GitHub's built-in notification mechanisms. At a convenient time, usually within a couple of days, one of the organizers processes the submission based on a series of scripts that invert the submission packaging process described above. Details are as follows:

(1) An unpack script uses our private key to decrypt the symmetric key used by the participant to encrypt the submission. The symmetric key is used to decrypt the tarball containing the runs and the metadata. The plaintext tarball is then unpacked, recovering the original (compressed) run files.

(2) The same `run_eval` script used by the participant is executed to evaluate both the development set queries as well as the evaluation set queries.

(3) The evaluation results are posted as a comment on the pull request by our GitHub service account called `msmarco-bot`. Along with the scores on the development and evaluation sets, the comment also provides some diagnostic information. The pull request is then merged into the main trunk of the repository, thus committing the run permanently. Note that the contents remain ciphertext; the plaintext is only transient for the purposes of evaluation and is removed after the evaluation scripts have run.

(4) The leaderboard CSV is updated with an entry corresponding to the submission, and the leaderboard webpage is correspondingly updated automatically. Embargoed submissions are shown as "Anonymous" on the leaderboard (more details in the next section). Earlier in the history of the leaderboard, we would send a tweet from the official MS MARCO Twitter account announcing a new state-of-the-art submission. The goal was primarily to generate excitement around the leaderboards. However, as the leaderboards "matured" we stopped doing this to be more aligned with the spirit of coopetition.

Currently, the above procedure for processing a submission is largely automated, but the scripts are still triggered manually by one of the organizers. Although in principle the entire process can be automated—e.g., a persistent agent that continuously monitors incoming pull requests—we have decided against this fully automatic design to ensure that the submission policy is not violated. Many issues of policy compliance are nuanced decisions that require human judgment, and in some previous cases, involved extensive discussions among the organizers before coming to a resolution. We cover these issues in the next section.

3.3 Submission Policy

We have attempted to operationalize the key points raised in Section 2.2 into a coherent submission policy that we detail in this section. It is important to note that our policy did not emerge fully

formed with the creation of the leaderboards, but evolved gradually over time as we grappled with issues that emerged. The evolution of the submission policy is perhaps an inevitable side effect of the leaderboard as a sequential game, as past participant behavior changes future behavior.

Frequency of Submission. The evaluation sets for both leaderboards are meant to serve as blind held-out test data. To combat the potential "leakage" issues discussed in Section 2.2, we request that participants submit:

(1) No more than two runs in any given period of 30 days.
(2) No runs that differ by very small changes, such as different random seeds or different hyperparameters.

We specifically request that participants who wish to run ablation studies do so on the development set.

There is a large parameter space of the form "no more than x runs in y days", and the setting that we decided on represents an attempt to tackle the issues discussed in Section 2.2 while still allowing participants reasonable access to scores on the held-out evaluation data. The restrictions on submission frequency were not implemented at the time the leaderboards were created, and some early participants submitted runs more frequently than would have been permitted following the adoption of the above policy.

Our policy naturally begs the question of how we define a participant, and there are no clear-cut answers (after much internal debate). Typically, a participant refers to a team, represented by a single individual making the submission. We felt it reasonable to consider different groups within a large organization (e.g., the same company or university) as distinct teams. However, a simple heuristic such as "no overlap in membership between different participating teams" is inadequate: for one, we do not require a team to disclose all its members, which would be impractical for large group efforts, and thus there is no membership list on which to base a decision. Furthermore, we have observed cases where overlaps in membership might be completely reasonable—for example, a senior researcher who participates in multiple efforts that are largely independent. Decisions about these corner cases are ultimately made on an ad hoc basis after deliberations among the organizers, typically in consultation with the participants.

Participant Identity. We expect that the team description in the metadata file (see Section 3.1) clearly and unambiguously identifies the participant. At the very least, the team description should include the name of the organization (e.g., university or company) and the name of the group within that organization. Each submission is encouraged to explicitly list the individual contributors of the run, but we do recognize that this might be impractical for large efforts. Specifically, aliases such as a generic or nondescript name are not permitted in the team description. We adopt the simple heuristic that a web search for the team description should unambiguously identify the participant.

The leaderboard does not allow anonymous submissions where the identity of the participant is (permanently) hidden, but we do allow *embargoed* submissions, where the goal is to facilitate blind reviewing for publications. Embargoed submissions must still contain accurate team and model information in the metadata JSON file (per above), but the corresponding entry on the leaderboard is anonymized, that is, shown as "Anonymous".

By default, we allow an embargo period of up to nine months. That is, after nine months, the identity of the submission will be revealed on the leaderboard. Additional extensions to the embargo period based on exceptional circumstances can be discussed on a case-by-case basis. Submission of an embargoed run is denoted by the following additional field in the metadata JSON file:

```
"embargo_until": "yyyy/mm/dd"
```

where the date in `yyyy/mm/dd` format cannot be more than nine months past the submission date. Of course, participants are free to specify a shorter embargo period if desired.

This aspect of the submission policy has undergone considerable evolution since the beginning of the leaderboard. Early on, we were rather lax about participant identities, and the distinction between anonymous and embargoed submissions was not clear, both in our own minds and as communicated to the participants. Furthermore, the pull request–based submission infrastructure was only put in place some time after the creation of the leaderboard. As a result, there are runs where it is no longer possible to reconstruct the identities of the participants (more in Section 4). However, we are quite stringent today about our requirements for participant identities, and regularly request that submission metadata be updated to meet our expectations.

Metadata Updates. The metadata provided at the time of a run submission is meant to be permanent. However, we do allow "reasonable" updates to the metadata as long as they abide by the spirit of the leaderboard. Acceptable reasons to request a metadata change include adding links to a paper or a code repository, fixing typos, clarifying the description of a run, etc. However, we reserve the right to reject any changes. To update the metadata of a particular run, we ask that the participant encrypt a new metadata JSON file with the same symmetric key used in the original submission and send a pull request to update the file.

4 CURRENT STATUS

The MS MARCO passage ranking leaderboard and document ranking leaderboard are held in separate GitHub repositories. The passage ranking leaderboard is a lot older and was launched in November 2018. As of mid-April 2022, we have received and processed 161 submissions. The document ranking leaderboard was launched in August 2020 and as of mid-April 2022, we have received and processed 116 submissions.

It is worth noting that the infrastructure described in this paper, based on GitHub pull requests, was not present at the initial creation of the leaderboards. For the document ranking leaderboard, our infrastructure was deployed relatively early, in September 2020, and thus most submissions were based on the pull request system (all except for 21 submissions). For the passage ranking leaderboard, the infrastructure was installed relatively late, in August 2021. Thus, we only have 35 submissions via pull requests to date.

Prior to the deployment of our infrastructure, submissions were accepted over email in an ad hoc fashion. Participants either emailed the run files directly as attachments or included links to public locations where the runs could be downloaded. Metadata was directly relayed in the texts of the emails. One of the organizers would process the submission by hand, inform the participant of the official

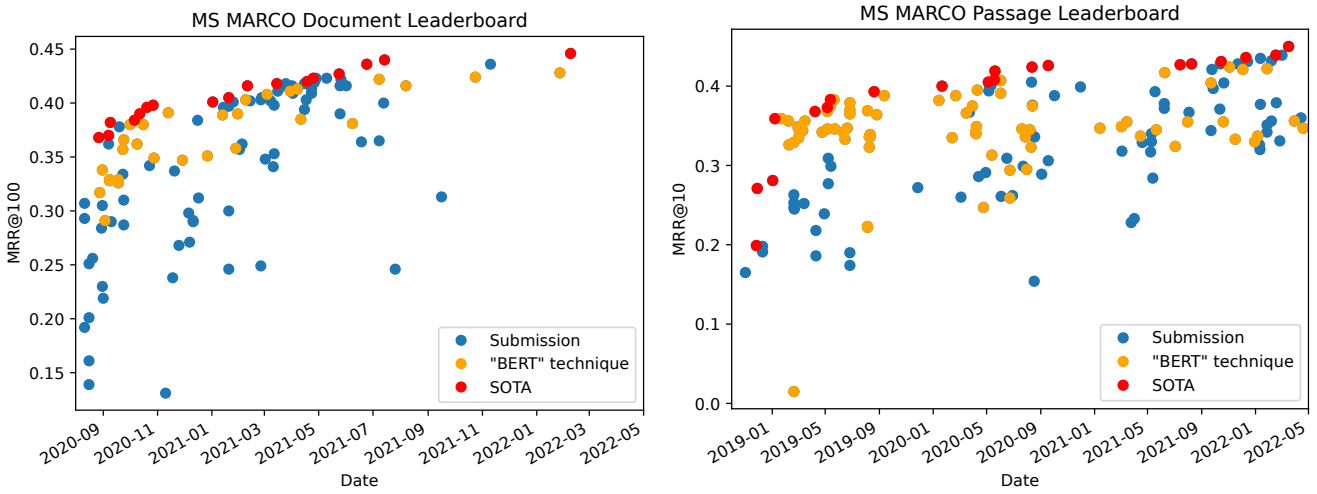


Figure 2: Overview of the MS MARCO document ranking leaderboard (left) and passage ranking leaderboard (right). Each point represents a submission, plotted with its submission date and effectiveness: orange points denote model descriptions that contain the string "BERT" and red points capture improvements in the "state of the art" over time.

results, update the leaderboard (which was an HTML table on a webpage), and then archive the runs on Microsoft file servers.

Unfortunately, this process was error prone, and the data management practices could have been better. For example, there was no consistent naming scheme for participants' run files and unambiguous association with metadata in our central archive. Furthermore, at least some of the submissions were received from what we would characterize as "burner" email accounts, created for the sole purpose of performing a submission; this relates to the probing behavior we discussed in Section 2.2. Furthermore, we did not clearly articulate a policy with respect to anonymous vs. embargoed runs until much later. The upshot is that there are entries on the passage leaderboard for which we do not have the associated run files and there are anonymous entries where it is impossible to reconstruct the participants' identities. Thus, conducting large-scale meta-evaluations and post-hoc analyses, such as those described in Craswell et al. [7], was more challenging than expected.

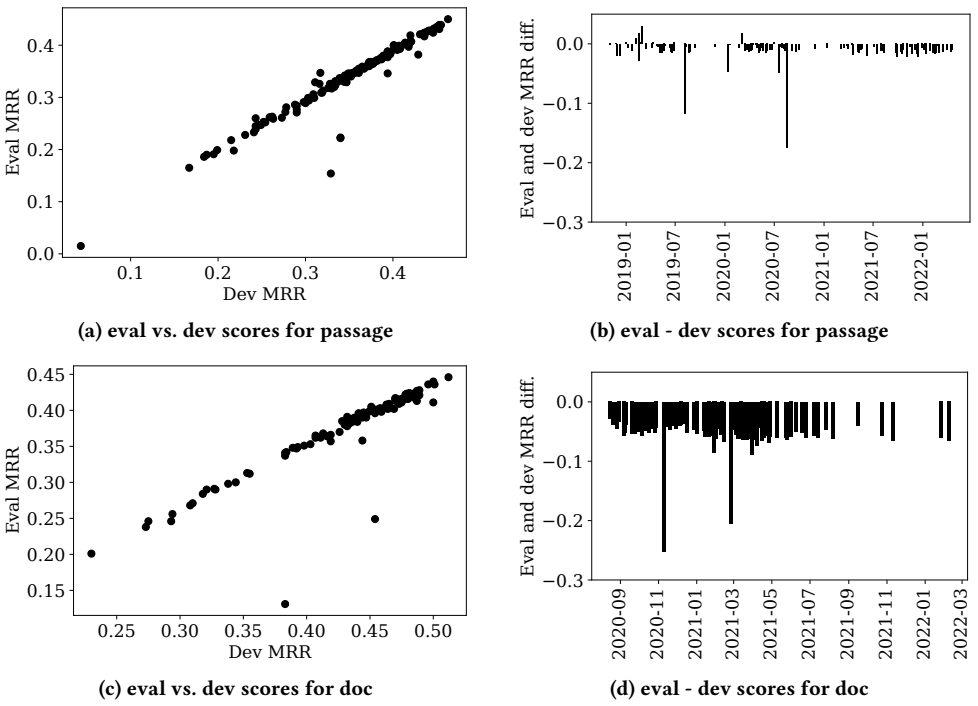
These issues were resolved with the implementation of the infrastructure described in this paper. We now have consistent naming of all run files, unambiguous associations between the run files and the metadata, and everything is in a machine-readable format. This is especially the case for the document leaderboard: because the infrastructure was deployed early, we have a high quality historical archive of submissions. Meta-evaluations and other analyses can now be easily scripted.

A summary of the two leaderboards from their launch until mid-April 2022 is shown in Figure 2: document ranking on the left and passage ranking on the right. In both plots, each point represents a run: the x-axis denotes the submission date and the y-axis shows the official metric on the held-out evaluation set (MRR@100 for document ranking, MRR@10 for passage ranking). The red dots denote (current and former) state-of-the-art (SOTA) runs atop each leaderboard. We see a gradual increase in the best scores over time, but see additional discussions in Lin et al. [12] and Craswell et al. [7] for more nuanced discussions of "SOTA".

As an illustration of the dominance of models based on BERT [10] in the leaderboard submissions, in both plots the orange dots represent model descriptions that contain the string "BERT" (not including runs that are currently embargoed). This captures a lower bound on the prevalence of methods based on pretrained transformers, as there are many runs that are clearly derived from BERT but do not contain "BERT" in its name, e.g., ANCE [18], which is popular on the document ranking leaderboard.

One obvious question is: Have our efforts been successful at addressing the "leakage" concerns central to this paper? Of course, it is not possible to assign causality based on the policy described in Section 3.3, but Craswell et al. [7] found that the MS MARCO leaderboard rankings are stable under a bootstrapping analysis, which provides evidence supporting the internal validity of the evaluations. This alleviates some of the concerns about participants overfitting to the evaluation set, although such checks should be conducted periodically to reaffirm the same in the future.

It is also interesting to consider if participants are overfitting to the development set. Figure 3 shows that there is a strong correlation between effectiveness on the evaluation and development sets for both the passage and document ranking tasks, shown in panels (a) and (c). In panels (b) and (d), we plot the gap between scores on the evaluation and development sets. The negative bars indicate that, with a few exceptions, scores on the evaluation sets are lower than those on the development sets, for both passage ranking and document ranking. In the case of document ranking, we observe that the gap has been increasing moderately over time, which may indicate some overfitting to the development set. While this analysis alone does not prove one way or another the effectiveness of our "leakage containment" efforts, there is nothing that jumps out at us as being potentially problematic or concerning.



(a) eval vs. dev scores for passage

(b) eval - dev scores for passage

(c) eval vs. dev scores for doc

(d) eval - dev scores for doc

Figure 3: Comparisons of effectiveness on the dev/eval sets for passage ranking (top panels) and document ranking (bottom panels). We show eval/dev scores in scatterplots (left panels) and the divergence between the scores over time (right panels).

5 REFLECTIONS

In this section, we reflect on some of the decisions we made in the creation and "maintenance" of the leaderboard over the past few years. We also share some plans that have yet to come to fruition.

Evolving Policy. The implementation, policy, and associated procedural mechanisms of the leaderboards did not emerge fully formed at the outset, but evolved gradually over time. This is perhaps not a surprise given the sequential nature of the leaderboard—participants adapt their behavior based on the actions of all other participants, and it make sense that organizers must adapt as well. We operated, for better or for worse, primarily in a reactive mode—issues came up, we discussed extensively, and made decisions, which are captured and organized in this paper. Examples include the frequency of submissions, participant identities, and anonymous vs. embargoed submissions. The result is that "rules" have not been consistently applied throughout the life of the leaderboard, e.g., nondescript team names in early submissions, more than two submissions within a 30-day window, etc.

Nevertheless, the issues that we have grappled with are not specific to MS MARCO, but are rather common across all leaderboards. It is our hope that the discussions captured in this paper can be of value to organizers of future leaderboards, so that they can do a better job of "getting it right" from the beginning.

Leaderboard Extensions. There are two ideas that we have discussed for quite some time, but have yet to come to fruition. These are worth sharing with the community.

The first is what we have called a "Docker" condition. Instead of submitting run files, a participant would submit a Docker image that generates the runs (i.e., in the pull request). Such an initiative would substantially advance the cause of reproducibility in the community. The technical infrastructure necessary to support such an evaluation strategy was already deployed for a SIGIR 2019 workshop [5], and in fact, a prototype framework for the MS MARCO document ranking leaderboard has already been implemented.

Unfortunately, this initiative never gained traction due to lack of interest from the community. Based on informal communications with participants at TREC, we received the feedback that such an idea was good "in the abstract", but would not be prioritized by researchers to actually contribute. Furthermore, we imagined that at least some participants (particularly those from industry) would have reservations about releasing Docker images that may contain proprietary data, models, and code. Since it would be unrealistic to allow *only* Docker-based submissions, we had no plans to eliminate file-based submissions. Thus, it was unclear what incentive participants would have for their extra effort. Our tentative answer was to set up a "private leaderboard" with additional queries (including potentially Microsoft internal data) that could only be accessed via a Docker image, but there were a host of logistical challenges associated with this idea, e.g., where would the hardware resources, queries, and relevance judgments come from?

Furthermore, easily reproducible runs would exacerbate the "leakage" issues we discussed in Section 2.2 and make it trivial for participants to perform model selection based on held-out evaluation scores. Due to a lack of enthusiasm from the community, at

least based on our perception, the "Docker condition" was never created, and unfortunately, we do not anticipate the environment changing in the near future to support such an initiative.

The second idea that we have discussed, but have yet to implement, is refreshing the held-out evaluation set periodically. For example, we could convert a single leaderboard into a sequence of leaderboards that "reset", for example, every quarter with a completely new held-out evaluation set. This structure might offer the best of both worlds—the rapid feedback afforded by leaderboards coupled with better methodological safeguards. We see no fundamental issues with such an approach, and the hurdles are only in terms of organizational logistics and availability of resources.

Ethical Considerations. Leaderboard-based evaluations can be useful to communities of researchers in providing a flexible mechanism to robustly benchmark model effectiveness on a shared task. In this way, they can aid rapid progress in model development and scientific knowledge production, assuming safeguards are in place to ensure that reliable conclusions can be drawn from the rankings. In the case of the MS MARCO leaderboards, in spite of our emphasis on coopetition and the policy detailed in Section 3.3, there is a risk that the leaderboard encourages SOTA-chasing, which has received valid critiques in the literature [2, 4, 16].

Craswell et al. [7] argued that leaderboards incentivize the community to work on specific problems and therefore, besides enforcing good scientific practices, organizers also bear the responsibility to be thoughtful about the impact of "funneling" a significant portion of the research community's attention to a small set of target tasks. In the case of shared tasks and leaderboards that have been introduced by industry, as is the case for MS MARCO, this deserves especially careful reflection on the impact of concentrating the work of task definitions in the hands of a few elite institutions [11] and the influence of industry in shaping academic research agenda [17]. Both the MS MARCO leaderboards and the TREC Deep Learning Tracks were designed to encourage exploration of data-hungry neural models, which may have the effect of "crowding out" alternative approaches that may be less methodologically suited to the current task guidelines.

Furthermore, the current MS MARCO leaderboards evaluate models *solely* on *mean* retrieval effectiveness but fail to consider the ecological and social costs [2], which are critical for meaningful comparisons. Specifically, issues that we have discussed internally include bias and fairness in retrieval results, as well as model size, model training cost, query latency, and other efficiency considerations. While there is nothing to prevent these considerations from being incorporated into leaderboard metrics, we have yet to do so for MS MARCO. This remains important future work.

To summarize: leaderboards, especially successful ones such as ours, hold great power in shaping the community and therefore we believe that beyond scientific considerations, leaderboard organizers must be conscious of ethical considerations and other externalities of their design choices as well.

6 CONCLUSIONS

The launch of the MS MARCO passage ranking leaderboard in November 2018 was roughly contemporaneous with the advent of BERT [10]. One of the biggest developments in information retrieval in recent memory—the application of pretrained transformers to ranking—was publicly demonstrated on the MS MARCO passage ranking leaderboard by Nogueira and Cho [15] with their monoBERT model in January 2019. This spurred a long run of rapid advances that continue to this day [13]. It would not be unfair to claim that the MS MARCO datasets in general and the leaderboards in particular have been instrumental in this progress. More than three years later, interest in the leaderboards remains healthy—and we look forward to further advances enabled by our efforts.

REFERENCES

[1] Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, Mir Rosenberg, Xia Song, Alina Stoica, Saurabh Tiwary, and Tong Wang. 2018. MS MARCO: A Human Generated MAchine Reading COmprehension Dataset. *arXiv:1611.09268v3* (2018).

[2] Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. On the Dangers of Stochastic Parrots: Can Language Models Be Too Big?. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency (FAccT '21)*. 610–623.

[3] Andrei Broder. 2002. A Taxonomy of Web Search. *SIGIR Forum* 36, 2 (2002), 3–10.

[4] Kenneth Ward Church and Valia Kordoni. 2022. Emerging Trends: SOTA-Chasing. *Natural Language Engineering* 28, 2 (2022), 249–269.

[5] Ryan Clancy, Nicola Ferro, Claudia Hauff, Jimmy Lin, Tetsuya Sakai, and Ze Zhong Wu. 2019. The SIGIR 2019 Open-Source IR Replicability Challenge (OSIRRC 2019). In *Proceedings of the 42nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2019)*. Paris, France, 1432–1434.

[6] Matt Crane. 2018. Questionable Answers in Question Answering Research: Reproducibility and Variability of Published Results. *Transactions of the Association for Computational Linguistics* 6 (2018), 241–252.

[7] Nick Craswell, Bhaskar Mitra, Daniel Campos, Emine Yilmaz, and Jimmy Lin. 2021. MS MARCO: Benchmarking Ranking Models in the Large-Data Regime. In *Proceedings of the 44th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2021)*. 1566–1576.

[8] Nick Craswell, Bhaskar Mitra, Emine Yilmaz, and Daniel Campos. 2020. Overview of the TREC 2020 Deep Learning Track. In *Proceedings of the Twenty-Ninth Text REtrieval Conference Proceedings (TREC 2020)*. Gaithersburg, Maryland.

[9] Nick Craswell, Bhaskar Mitra, Emine Yilmaz, Daniel Campos, and Ellen M. Voorhees. 2019. Overview of the TREC 2019 Deep Learning Track. In *Proceedings of the Twenty-Eighth Text REtrieval Conference Proceedings (TREC 2019)*. Gaithersburg, Maryland.

[10] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota, 4171–4186.

[11] Bernard Koch, Emily Denton, Alex Hanna, and Jacob G. Foster. 2021. Reduced, Reused and Recycled: The Life of a Dataset in Machine Learning Research. *arXiv:2112.01716* (2021).

[12] Jimmy Lin, Daniel Campos, Nick Craswell, Bhaskar Mitra, and Emine Yilmaz. 2021. Significant Improvements over the State of the Art? A Case Study of the MS MARCO Document Ranking Leaderboard. In *Proceedings of the 44th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2021)*. 2283–2287.

[13] Jimmy Lin, Rodrigo Nogueira, and Andrew Yates. 2021. *Pretrained Transformers for Text Ranking: BERT and Beyond.* Morgan & Claypool Publishers.

[14] Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. MS MARCO: A Human Generated MAchine Reading COmprehension Dataset. *arXiv:1611.09268v1* (2016).

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

[16] Inioluwa Deborah Raji, Emily M. Bender, Amandalynne Paullada, Emily Denton, and Alex Hanna. 2021. AI and the Everything in the Whole Wide World Benchmark. *arXiv:2111.15366* (2021).

[17] Meredith Whittaker. 2021. The Steep Cost of Capture. *Interactions* 28, 6 (2021), 50–55.

[18] 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 *Proceedings of the 9th International Conference on Learning Representations (ICLR 2021)*.