Proceedings of the Seventh International Workshop on Evaluating Information Access (EVIA 2016),
a Satellite Workshop of the NTCIR-12 Conference, June 7, 2016 Tokyo Japan

Promoting Repeatability Through Open Runs

Ellen M. Voorhees, Shahzad Rajput, Ian Soboroff
National Institute of Standards and Technology*
Gaithersburg, MD

ABSTRACT

The 2015 Text REtrieval Conference (TREC) introduced the concept of 'Open Runs' in response to the increasing focus on repeatability of information retrieval experiments. An Open Run is a TREC submission backed by a software repository such that the software in the repository reproduces the system that created that exact run. The ID of the repository was captured during the process of submitting the run and published as part of the metadata describing the run in the TREC proceedings. Submitting a run as an Open Run was optional: either a repository ID was provided at submission time or it was not, and further processing of the run was identical in either case. Unfortunately, this initial offering was not successful. While a healthy 79 runs were submitted as Open Runs, we could not in fact reproduce any of them. This paper explores possible reasons for the difficulties and makes suggestions for how to address the deficiencies so as to strengthen the Open Run program for TREC 2016.

1. INTRODUCTION

Experimentation is a fundamental component of science, and verification of an experimental result by an independent party—reproducibility—an established tenet of good experimental practice. Recently, critics have raised the alarm about the lack of reproducibility of published results especially in the life sciences [5]. The critics have listed a number of factors that contribute to the lack of reproducibility ranging from lack of incentives to reproduce another's experiment to over-emphasis on p-values as an indication of Truth.

Information retrieval (IR) research has long been an experimental discipline, and the concerns regarding reproducibility for science in general have also been reflected in concerns for retrieval research more particularly. In response, the ECIR 2015 conference solicited papers that reproduced other studies and three such papers were presented at the conference [3, 4, 6]. SIGIR 2015 hosted the Workshop on Reproducibility, Inexplicability and Generalizability Of Results (RIGOR) [1]. That workshop distinguished between *repeatability* and *reproducibility* of IR results. It defined repeatability as the ability "to repeat a published result under approximately the same conditions in which the previously published experiments occurred", and defined reproducibility as the ability to "reproduce a published result on a comparable dataset".

To promote repeatability of TREC[1] experiments, TREC 2015 introduced the idea of an Open Run, a TREC submission backed by a software repository such as on GitHub[2] that captures the code to recreate the run. The initiative has similar goals as the ACM SIGMOD Reproducibility effort (db-reproducibility.seas.harvard.edu), though there are differences in how the two programs are implemented. In 2015, TREC made no effort to vet repositories, but accepted whatever repository ID the TREC participant provided. In contrast, authors of papers accepted to the SIGMOD conference have the option of working with a member of the Reproducibility Committee to assist that member in reproducing the main results reported in the paper. The data and scripts used by the Reproducibility Committee member become the publicly-available artifacts associated with the paper.

The Open Run initiative garnered reasonable support with 79 runs submitted as Open Runs, but, unfortunately, we were unable to actually recreate the submission for any Open Run. This paper examines this first year of the initiative so as to strengthen the program for TREC 2016. The next section describes how the Open Runs process worked in TREC 2015, while Section 3 lists the lessons learned from the effort.

2. IMPLEMENTATION IN TREC 2015

The instructions regarding Open Runs were contained in the Welcome message that is the first message sent to TREC participants in a given year. The entire discussion of Open Runs was contained in the four paragraphs shown in Figure 1. In addition, the run submission form contained a field labeled 'OPEN RUN URL' whose instructions reminded TREC participants about the purpose of Open Runs and pointed them back to the Welcome message:

> In support of repeatability of TREC experiments, TREC 2015 is introducing the concept of Open Runs. An Open Run is a TREC run

*Certain commercial equipment, instruments, or materials are identified in this paper in order to specify the experimental procedure adequately. Such identification is not intended to imply recommendation or endorsement by the National Institute of Standards and Technology, nor is it intended to imply that the materials or equipment identified are necessarily the best available for the purpose.

[1] trec.nist.gov
[2] github.com

To support the repeatability of TREC experiments, we are introducing the concept of Open Runs. An Open Run is a TREC run submission backed by a GitHub repository. GitHub is a source code revision control hosting site; you can host your code there, publish changes and releases, and anyone can access that code using either the website or the open-source git version control tool. If you submit your run as an Open Run, anyone else can reproduce your results by cloning your GitHub repository.

The procedure for submitting an Open Run is as follows. Your code must be in a GitHub repository, and when you are ready to submit your run, you must tag the code with a 'TREC2015-submission' tag. In the repository there needs to be documentation and scripts so that someone cloning the repo can reproduce the run; we suggest a README.TREC in a docs/ folder, and a script called 'generate-TREC2015-run'. Do not include data or topics in your repository.

When you submit your run, there will be a place on the submission form for you to give the URL for an open run. Here you should paste the URL pointing to the tagged release of your code, for example, 'https://github.com/example/tree/TREC2015-submission'. You can get this URL from the GitHub website by selecting the tag from the tags/branches drop down menu. This URL will be published with the final TREC results in the run archive, so that once the TREC cycle is complete others can reproduce your results.

Open Runs are an opportunity not only to promote reproducibility of TREC results, but also to benefit other potential participants by publishing systems that can be used as baselines in future years. By making your runs Open, you contribute to the community. Open Runs are not mandatory, as releasing source code for your system may not be possible, but we hope that you will consider making your runs Open if you can.

Figure 1: Instructions to TREC participants regarding the new Open Run initiative.

Table 1: Number of teams participating in a track and number of teams that submitted their runs as Open Runs to that track for TREC 2015 tracks attracting any Open Runs.

Track	Total Teams	Teams Submitting Open Runs
Clinical Decision Support	36	8
Contextual Suggestion	12	2
Microblog	16	7
Tasks	5	2

> submission backed by a GitHub repository—specifically, a GitHub repository that encapsulates all the code needed to produce this exact run. See the TREC 2015 welcome message for more details. Provide the URL for the tagged release of the repository that reproduces this run here.

TREC 2015 received a total of 79 individual run submissions that included some sort of ID in the Open Run field. Multiple runs from the same participating team to the same track were always included in the same repository, resulting in a total of 19 distinct repositories. There were 87 participating teams in TREC 2015, so slightly less than one quarter of the teams submitted their runs as Open Runs. Table 1 shows the number of teams submitting Open Runs, as well as the total number of teams participating in the track, for each of the four tracks that received any Open Runs.

In December 2015, the conclusion of the TREC 2015 cycle, we accessed each repository to try and recreate the submissions. We made a good faith effort to recreate a submission from the contents of its repository, though we purposely did not go to extraordinary lengths to do a reconstruction since the target audience of the repositories are unlikely to do so. In the end, we were unable to reproduce any of the runs due to a few different types of difficulties.

Some repositories were either invalid, i.e, the URL provided did not point to a repository, or empty, i.e., the URL pointed to a repository but that repository contained no files. One repository contained only code that the track organizers had released; there was no code specific to the participant who had created the repository. Other repositories contained code but no README file or other indication of how to build the system. Few users will be motivated enough to trace through a mass of undocumented code to understand how to build the system and create a run (and we did not). In another few repositories we began to build the system but encountered compiler errors during the build. For one repository, the system built cleanly, but reproducing the TREC submission required proprietary data that was not available to us. Table 2 shows the counts of the repositories affected by a given difficulty. The number of repositories sums to more than 19 (the total number of repositories) because we could encounter more than one difficulty per repository.

Table 2: Number of Open Run repositories affected by a given difficulty.

Invalid or empty repository:	4
Repository contains no run-specific code:	1
Missing/incomplete README file:	9
Compilation error encountered:	5
Proprietary data needed:	1

3. DISCUSSION

Our initial experience with Open Runs has been instructive despite our inability to reproduce any runs. In this section we outline some of the lessons learned.

3.1 Participation

An almost 25% participation rate for a new initiative is actually encouraging, especially given that TREC's definition of Open Runs requires a very high degree of transparency—the entire retrieval system is assumed to be open source.

This level of transparency is too great for many participants who have legitimate proprietary interest in their systems. In the future TREC may wish to ratchet down the level of transparency required. For example, a repository containing a complete specification file for a run, but not the retrieval system code itself, will be more useful than no repository at all.

Having said that, it is unclear how best to motivate participants to submit Open Runs. Creating an effective repository requires a significant amount of additional work (documenting and packaging the experiment) compared to only submitting the run. Further, this work comes at a time when the participant is facing deadline pressures. We assume that the empty and invalid repositories observed in the TREC 2015 submissions reflect participants' intentions to make the run open once the deadline pressure was gone, intentions that unfortunately were not fulfilled. Since the rewards for submitting Open Runs are intangibles—essentially, being a good citizen of the research community—branding a run as an open run and pointing to the associated repository appear to be the only available incentives to offer to TREC teams to encourage them to participate in the Open Runs program. Similarly, a paper that goes through the SIGMOD Reproducibility process gets a "db-reproducible" tag attached to it in the ACM digital library as its reward.

3.2 Data

As with transparency, the issue of data within a repository is also more nuanced than the TREC definition of Open Runs affords. In the TREC 2015 instructions, the explicit prohibition against including data in the repository was intended to prevent participants from including the common track data (documents and topics) in the repository since anyone recreating the run would presumably have that data, too. But, of course, retrieval systems might use all sorts of auxiliary resources in processing a run, such as thesauri or other vocabulary aids, corpora other than the target document set, previously learned models of items of interest, etc. Some of these resources might be able to be included in the repository itself and others specified at a sufficient level of detail that an interested repository user could procure the appropriate version. But some will not be transferable at all. For example, a system that builds queries based on relationships it mines from the live web at search time will likely create different queries every time it is run.

Auxiliary data resources are not the only data challenges. TREC tasks that employ dynamic data streams also violate the basic premise of Open Runs—that it is possible to capture the software of a TREC participant at the point of time when a TREC submission was created and use that exact code base to recreate a run. In TREC 2015, The Microblog track was one of the tracks that used a live data stream as the document set. Participants monitored the live Twitter[3] stream to find tweets relevant to a set of (simulated) user profiles. Participants' systems necessarily contain routines that connect to Twitter and process the observed tweets. Since the tweet stream is constantly changing, rerunning that code definitely does not recreate the submitted run.

Open Runs as defined also exclude the variety of information seeking tasks that do not fit neatly into the test collection framework. A prime example in this category is interactive runs. In an interactive run, the behavior of the system is dependent on the particular choices made by the user. Since different users will interact differently with the system, and the same user will interact differently at different times, directly recreating the submission will not be possible unless some sort of trace of the original interaction is recorded and used to simulate the user during the recreation [2].

3.3 Instructions

The rudimentary definition of Open Runs was not a main cause of our inability to recreate the runs that were submitted as Open Runs in TREC 2015. A sizeable majority of the difficulties were much more prosaic: lack of a README file or other explanation of the way forward and compilation errors.

The instructions sent to TREC participants gave little direction as to how best to set up a usable repository. More extensive instructions may reduce the number of prosaic errors. In particular, the instructions should stress the importance of a README file and encourage the use of associated 'make' files to guide a repository user through the process of building the system and creating a run. The README file should document the computing environment (operating system, compiler version, etc.) in which the submission was created. Any other known dependencies should also be explicitly listed in the README. A set of example repositories, one per track, containing a baseline or stub system for accomplishing the track's task and complete documentation in the desired format would make expectations clear and assist participants in creating their own repositories. ReproZip[4] is another option to ease the construction of effective repositories.

3.4 Next Steps

The Open Runs initiative will continue in TREC 2016. In light of the above considerations, we propose some modifications to the program.

- **Separate the deadlines for run submission and Open Run designation**. If our assumption that empty/invalid repositories reflect participants' intention to submit open runs that was thwarted by deadline pressure is true, then a later deadline for designating a run as an Open Run should improve the quality of the repositories.

- **Validate the repositories**. A process whereby the repositories are tested within the TREC cycle would allow errors to be caught at a time when they could still be fixed. For example, participants in a track could try reproducing one anothers' runs from the repositories. Since peer review of repositories might prove to be unduly burdensome, an alternative is to have a team reproduce its own result from the repository on a separate machine. Documentation of such an effort could be part of the Open Run designation submission.

- **Require (or at least strongly encourage) use of a standardized repository structure**. Use of a standardized repository structure would not only help the teams creating Open Runs by reminding

[3] twitter.com

[4] reprozip.readthedocs.io/en/1.0.x/

them of the items a good repository contains, but will also facilitate use of the repositories by lowering the learning curve of how to recreate a run. But ease-of-repository-use must be balanced against ease-of-repository-creation since participating teams must remain willing to submit Open Runs.

We suggest that part of the standardized structure be explicit support for abandoning one-repository-per-run in favor of one-repository-per-participation-in-a-track. Each of the TREC 2015 teams did this despite the original instructions to create one repository per run. Building repositories at the track participation level is less work for the TREC participant and corresponds better to the desired outcome of repeatability of a retrieval experiment.

- **Support the use of data streams for real-time tasks**. For TREC tasks that use live streams of data as the document set, instruct participants to configure their system such that all code related to opening and reading from the stream is localized to a single module. Also, have someone (say, track organizers) capture the live stream in such a way that it can be replayed. (This replaying may not be able to be identical to the original stream.) Submissions to real-time tasks can then be (mostly) recreated by having the system attach to the replayed stream as opposed to the truly live stream.

4. CONCLUSION

The Open Runs initiative introduced in TREC 2015 has the goal of promoting reproducibility in IR research by recognizing TREC participants who support their submissions with the code that produces them. The first year demonstrated that there is interest in the program, but also that the initial TREC definition of Open Runs is simplistic. Many current retrieval systems use resources other than the target document set's text, and some have stochastic components. Search tasks may focus on real-time data streams or supporting human-in-the-loop interactions. In some of these situations it is simply not possible to reproduce the submission file verbatim. For all of them, a software-exclusive clone of the system that created the TREC submission is insufficient.

The Open Runs initiative will continue in TREC 2016 with a continued emphasis on the simple (and common) case where a software clone is sufficient. More detailed instructions on how to create a repository, and timely reminders sent to participants before deadline crunches, may help with the invalid repository and compilation error problems that affected the majority of 2015 repositories. We will also engage with the TREC community to develop a mechanism for specifying external resources required by the system.

5. ACKNOWLEDGEMENTS

The authors thank the reviewers for their helpful suggestions. We are also thankful for the intrepid TREC participants who were willing to be pioneers in the Open Run program.

6. REFERENCES

[1] J. Arguello, M. Crane, F. Diaz, J. Lin, and A. Trotman. Report on the SIGIR 2015 workshop on Reproducibility, Inexplicability, and Generalizability Of Results (RIGOR). *ACM SIGIR Forum*, 49(2):107–116, December 2015.

[2] L. Azzopardi, K. Järvelin, J. Kamps, and M. D. Smucker. Report on the SIGIR 2010 workshop on the simulation of interaction. *ACM SIGIR Forum*, 44(2):35–47, December 2010.

[3] N. Ferro and G. Silvello. Rank-biased precision reloaded: Reproducibility and generalization. In *Advances in Information Retrieval. Proceedings of the 37th European Conference on IR Research, ECIR 2015*, pages 768–780, 2015.

[4] M. Hagen, M. Potthast, M. Büchner, and B. Stein. Twitter sentitment detection via ensemble classification using averaged confidence scores. In *Advances in Information Retrieval. Proceedings of the 37th European Conference on IR Research, ECIR 2015*, pages 741–754, 2015.

[5] J. P. Ioannidis. Why most published research findings are false. *PLoS Med*, 2(8):e124, 2005. doi:10.1371/journal.pmed.0020124.

[6] J. Rao, J. Lin, and M. Efron. Reproducible experiments on lexical and temporal feedback for Tweet search. In *Advances in Information Retrieval. Proceedings of the 37th European Conference on IR Research, ECIR 2015*, pages 755–767, 2015.