TIREx: The Information Retrieval Experiment Platform

TIREx Integrates Existing Tools:
- TIRA
  - Reproducible shared tasks
  - Software submissions
  - Blinded experiments
- IR datasets
  - Unified data access
  - Documents + queries + qrels
- PyTerrier
  - Reproducibility pipelines

Shared Tasks with Software Submissions
Why?
- Reproducibility and replicability are long-standing problems
- We can not ensure that LLMs are not trained on test data

How?
- Organizer upload docker image with IR datasets integration
- Participants upload docker images with retrieval approaches
- Sandbox and blinded execution of immutable software
- Improves reproducibility
- Potentially confidential data

Towards Reproducible and Replicable Shared Tasks in IR via Software Submissions

Benefits for Organizers
- Approaches submitted to previous editions can be re-executed
- Diversification of pools for shared tasks with few participants
- Test data can remain private
- Integration to IR datasets increases the adoption of the dataset

Benefits for Post-Hoc Experiments
Repeat, replicate, and reproduce in one line of code.
Organizers of a shared task can publish the artifacts produced during the shared task as a git repository. Researchers can use the resulting shared task artifacts (data and submitted software) in their experiments.

Examples:
- Declarative PyTerrier pipeline for full-rank retrieval from a complete corpus.
  ```py
  pipeline = tira.pt.retriever('bm25', wmodel='BM25')
  reranker = tira.pt.reranker('BM25')
  )
  advanced_pipeline = pipeline >> advanced_reranker
  ```
- Declarative PyTerrier pipeline to re-rank BM25 with a submitted software.
  ```py
  bm25 = pt.BatchRetrieve(index, wmodel='BM25')
  reranker = bm25 >> tira.pt.reranker('bm25')
  )
  ```

Benefits for Participants
- One software submission, evaluation on many datasets
- Multi-stage pipelines are first-class citizens
  - Output of previous stages as additional input
  - Efficiency by Caching due to immutability of software
- Support for Re-Rankers
  - Unified data interface via IR datasets
  - Allows modularization: Chain arbitrary re-rankers
- Support for external APIs / manual annotations via data uploads

Feasibility Study: 50 Retrieval Models on 32 IR-Benchmarks

<table>
<thead>
<tr>
<th>Corpus</th>
<th>Docs.</th>
<th>Size</th>
<th>Tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td>ArgSme</td>
<td>0.4 m</td>
<td>8.3 GB</td>
<td>2</td>
</tr>
<tr>
<td>Antique</td>
<td>0.4 m</td>
<td>90.0 MB</td>
<td>1</td>
</tr>
<tr>
<td>ClueWeb09</td>
<td>1.0 b</td>
<td>40.0 TB</td>
<td>4</td>
</tr>
<tr>
<td>ClueWeb12</td>
<td>731.7 m</td>
<td>45.8 TB</td>
<td>4</td>
</tr>
<tr>
<td>ClueWeb22B</td>
<td>200.0 m</td>
<td>6.8 TB</td>
<td>1</td>
</tr>
<tr>
<td>CORD-19</td>
<td>0.2 m</td>
<td>7.1 GB</td>
<td>1</td>
</tr>
<tr>
<td>Cranfield</td>
<td>1,400</td>
<td>0.5 MB</td>
<td>1</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>WaPo</td>
<td>0.6 m</td>
<td>1.6 GB</td>
<td>1</td>
</tr>
<tr>
<td>= 15 corpora</td>
<td>1.9 b</td>
<td>15.3 TB</td>
<td>32</td>
</tr>
</tbody>
</table>

To fill the leaderboards, we executed all 50 models on all 32 benchmarks.

Teaser Experiment Results:
We observe system preferences on TREC DL 2019 and measure the proportion of reproducible preferences with repro.eval.

<table>
<thead>
<tr>
<th>Task</th>
<th>Rank</th>
<th>Success</th>
</tr>
</thead>
<tbody>
<tr>
<td>TREC DL 2020</td>
<td>1</td>
<td>88.1</td>
</tr>
<tr>
<td>Core 2018</td>
<td>5</td>
<td>70.2</td>
</tr>
<tr>
<td>Web track 2003</td>
<td>15</td>
<td>57.8</td>
</tr>
<tr>
<td>Web track 2013</td>
<td>30</td>
<td>31.0</td>
</tr>
</tbody>
</table>

Your next Experiment?
Metadata and results from TIREx are valuable for future experiments: LTR, QPP, etc.
We would be happy to help you bring future experiments or shared tasks to TIREx!