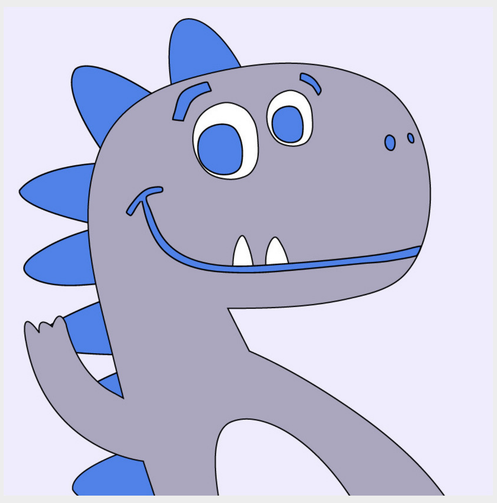
Maik Fröbe Jan Heinrich Reimer Sean MacAvaney Niklas Deckers Simon Reich Janek Bevendorff Benno Stein Matthias Hagen Martin Potthast
maik.froebe@uni-jena.de

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

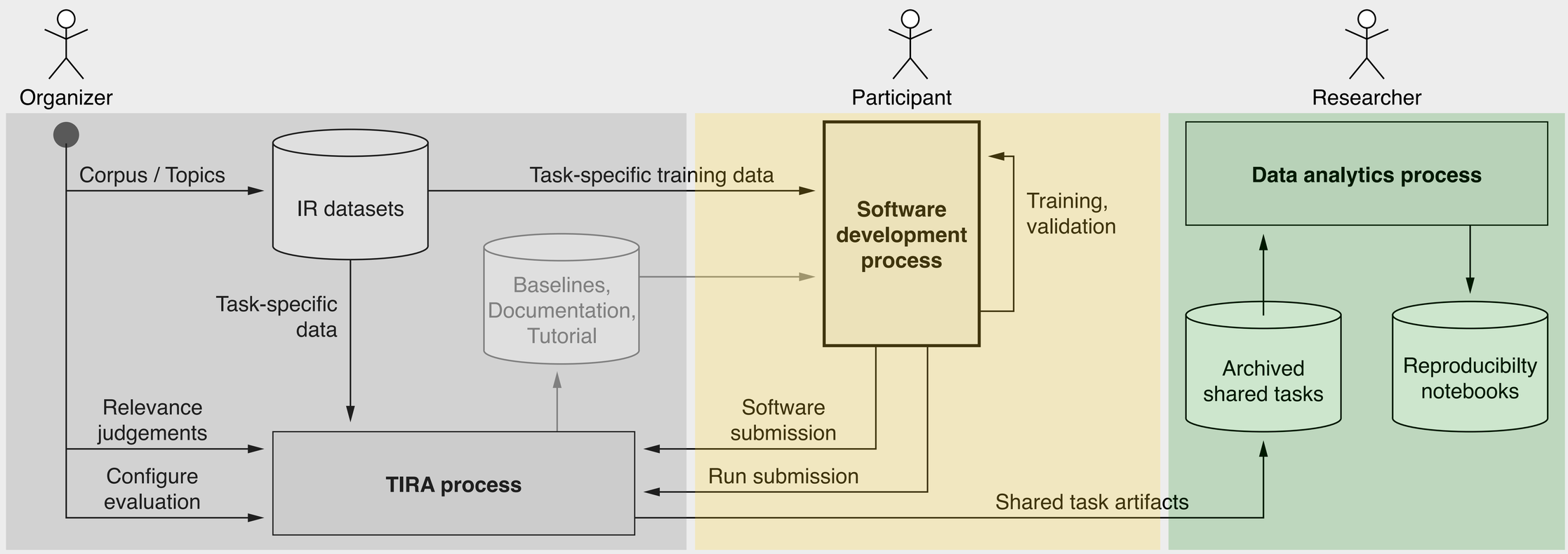
PyTerrier
- Reproducibility pipelines

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
- Sandboxed 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 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

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.

```
pipeline = tira.pt.retriever(
  '<task-name>/<user-name>/<software>',
  dataset = '<dataset>'
)
advanced_pipeline = pipeline >> advanced_reranker
```

Declarative PyTerrier pipeline to re-rank BM25 with a submitted software.

```
bm25 = pt.BatchRetrieve(index,wmodel='BM25')
reranker = bm25 >> tira.pt.reranker(
  '<task-name>/<user-name>/<software>'
)
```

Feasibility Study: 50 Retrieval Models on 32 IR-Benchmarks

Corpus			Tasks
Name	Docs.	Size	
Args.me	0.4 m	8.3 GB	2
Antique	0.4 m	90.0 MB	1
ClueWeb09	1.0 b	4.0 TB	4
ClueWeb12	731.7 m	4.5 TB	4
ClueWeb22B	200.0 m	6.8 TB	1
CORD-19	0.2 m	7.1 GB	1
Cranfield	1,400	0.5 MB	1
	...		
WaPo	0.6 m	1.6 GB	1
Σ = 15 corpora	1.9 b	15.3 TB	32

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

Framework	Type	Approaches	
		Full-rank	Re-rank
BEIR	Bi-encoder	17	17
ChatNoir	BM25F	1	0
ColBERT@PT	Late interaction	0	1
DuoT5@PT	Cross-encoder	0	3
PyGaggle	Cross-encoder	0	8
PyTerrier	Lexical	20	20
Pyserini*	Lexical	4	4

Teaser Experiment Results:

We observe system preferences on TREC DL 2019 and measure the proportion of reprodicible preferences with repro_eval.

Task	Rank	Success
TREC DL 2020	1	88.1
Core 2018	5	70.2
Web track 2003	15	57.8
Web track 2013	30	31.0

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!