Improving the Reproducibility of PAN’s Shared Tasks

Bauhaus-Universität Weimar	Martin Potthast, Tim Gollub, Benno Stein
Universitat Politècnica de València	Paolo Rosso
Autoritas Consulting	Francisco Rangel
University of the Aegean	Efstathios Stamatatos

[pan.webis.de]

Improving the Reproducibility of PAN’s Shared Tasks

Outline

- About Shared Tasks
- The TIRA experiment platform
- Plagiarism Detection, Author Profling, and Author Identification
- Summary

About Shared Tasks

About Shared Tasks

Terminology

The term “shared task” refers to computer science events that invite researchers and practitioners to work on a specific problem of interest, the task.*

Goals

- ❑ development of new theories / approaches
- ❑ implementation of suited softwares
- ❑ evaluation of currently achievable performance

*Typical terms used in this regard are: campaign, challenge, competition, contest, or cup.

About Shared Tasks

Terminology

The term “shared task” refers to computer science events that invite researchers and practitioners to work on a specific problem of interest, the task.*

Goals

- development of new theories / approaches
- implementation of suited softwares
- evaluation of currently achievable performance

Pros

- task standardization
- evaluation resource development
- transfer from academia to industry

Cons

- “task concentration” (less diversity)
- winner imitation
- repeated participation fatigue

*Typical terms used in this regard are: campaign, challenge, competition, contest, or cup.

About Shared Tasks

Terminology

The term “shared task” refers to computer science events that invite researchers and practitioners to work on a specific problem of interest, the task.*

Goals

- development of new theories / approaches
- implementation of suited softwares
- evaluation of currently achievable performance

Pros

- task standardization
- evaluation resource development
- transfer from academia to industry

Cons

- “task concentration” (less diversity)
- winner imitation
- repeated participation fatigue

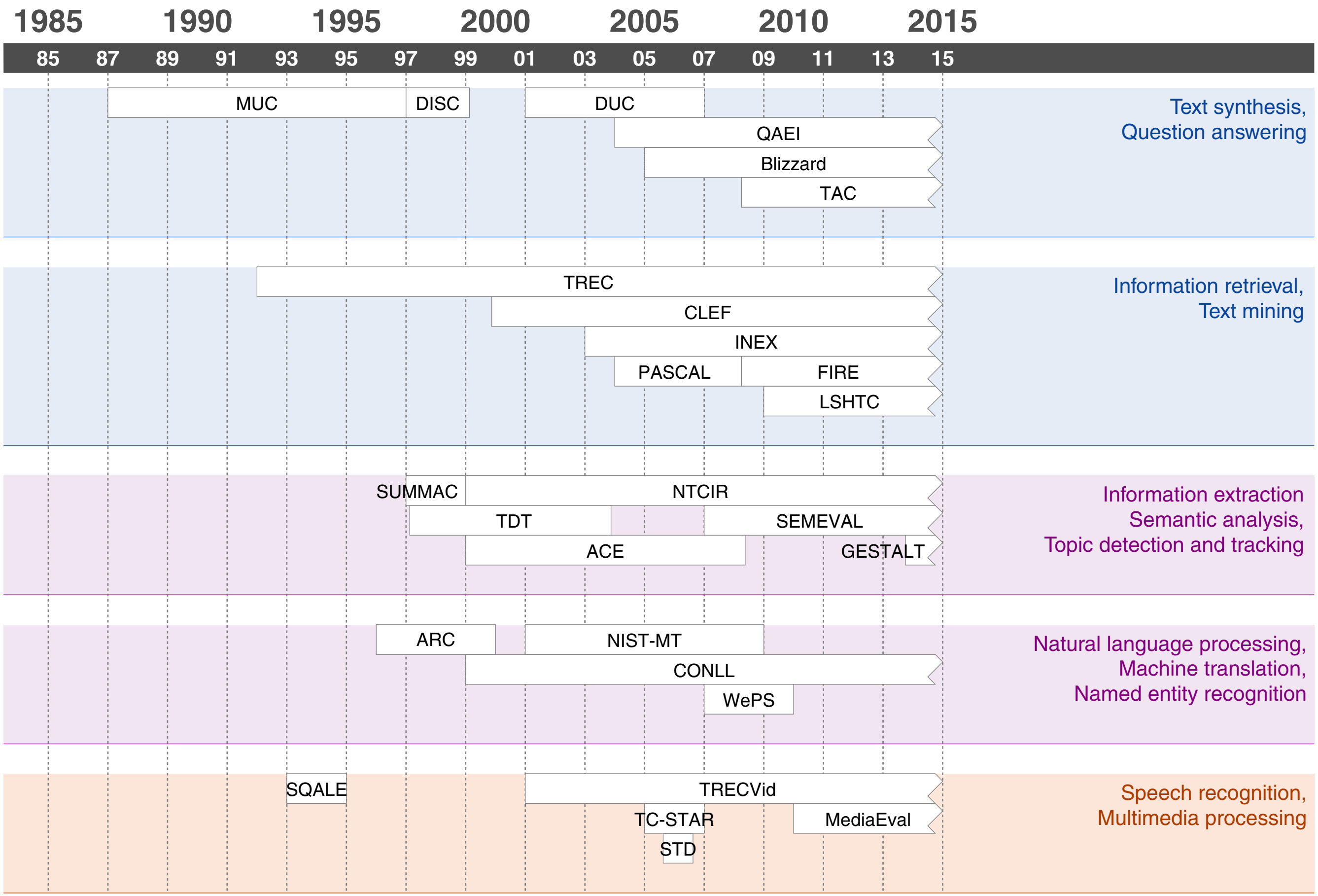
Success indicators

- participation (registrations, downloads, submissions)
- scientific impact (citations)

*Typical terms used in this regard are: campaign, challenge, competition, contest, or cup.

About Shared Tasks

Timeline of Shared Tasks in Human Language Technologies



About Shared Tasks

Shared Tasks by Submission Type

Corpus (and what may be published to participants)

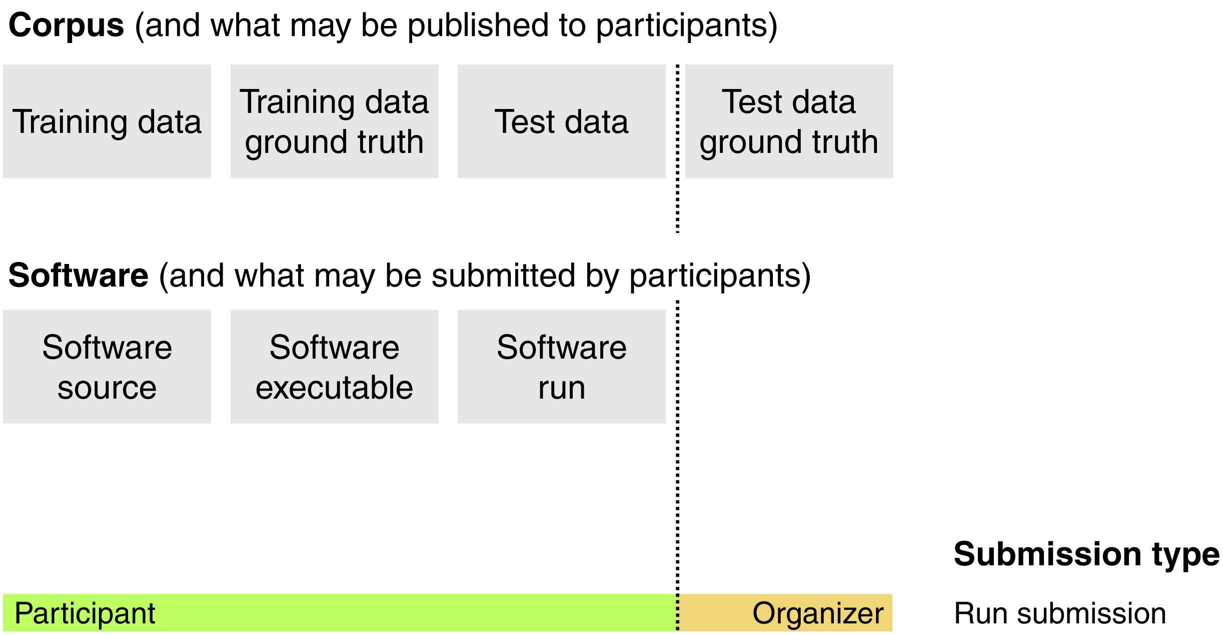
Training data	Training data ground truth	Test data	Test data ground truth

Software (and what may be submitted by participants)

Software source	Software executable	Software run

About Shared Tasks

Shared Tasks by Submission Type



Corpus (and what may be published to participants)

Training data

Training data ground truth

Test data

Test data ground truth

Software (and what may be submitted by participants)

Software source

Software executable

Software run

Submission type

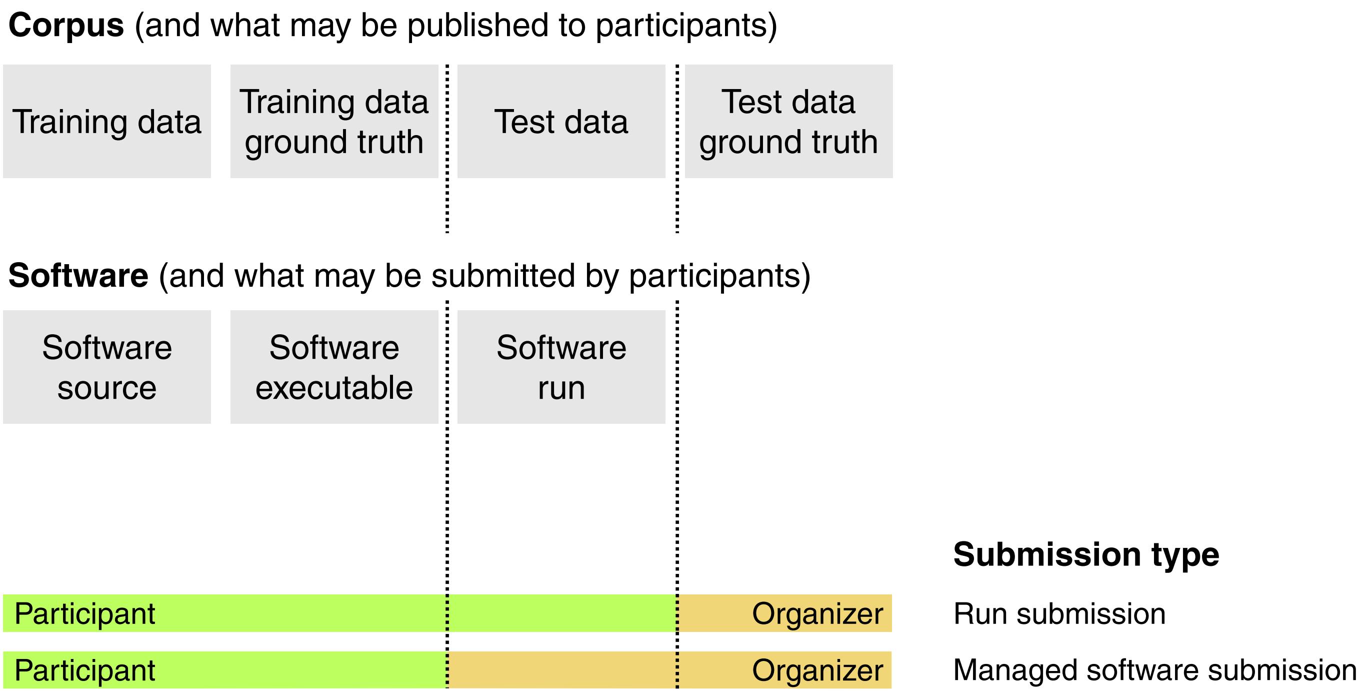
Participant

Organizer

Run submission

About Shared Tasks

Shared Tasks by Submission Type



Corpus (and what may be published to participants)

Training data

Training data ground truth

Test data

Test data ground truth

Software (and what may be submitted by participants)

Software source

Software executable

Software run

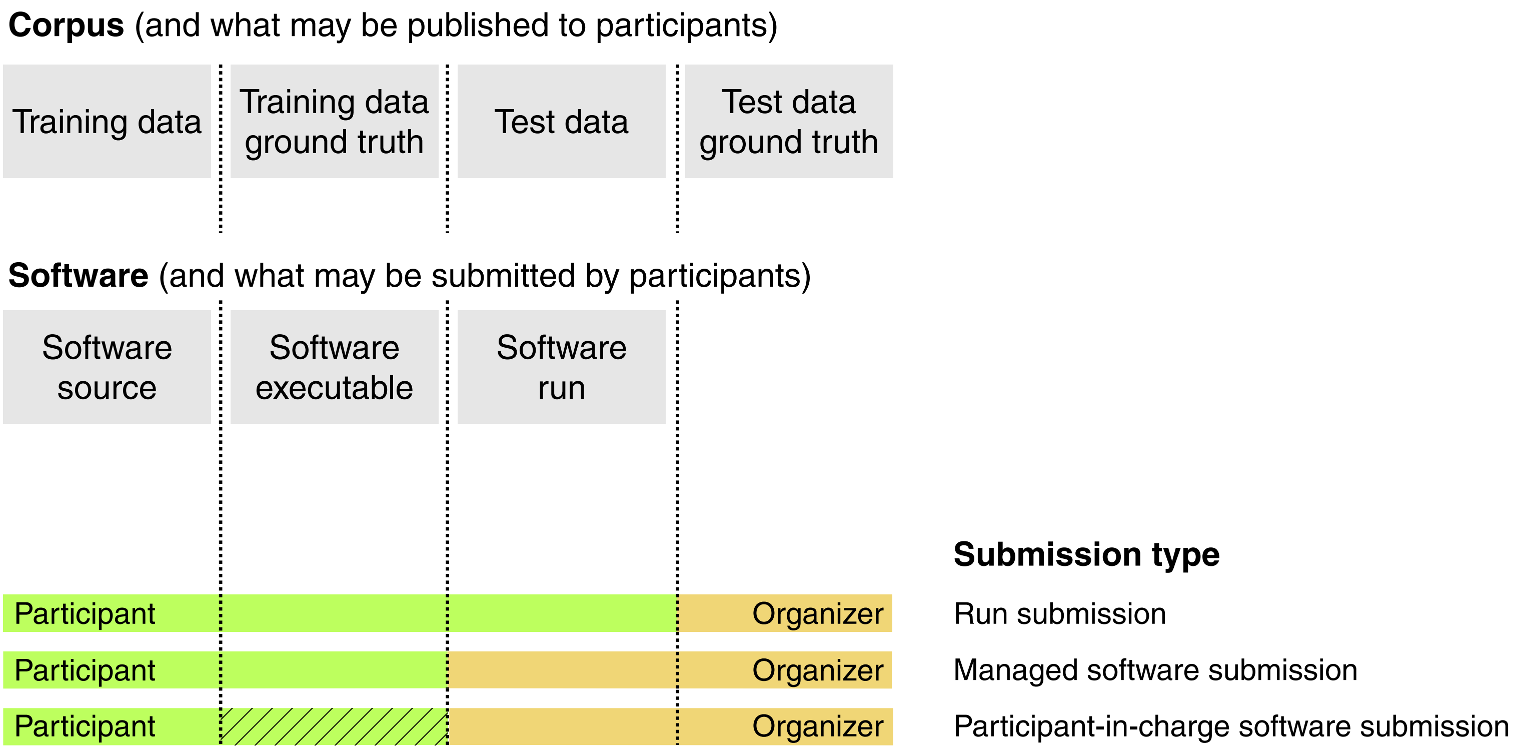
Submission type

Participant — Organizer: Run submission

Participant — Organizer: Managed software submission

About Shared Tasks

Shared Tasks by Submission Type



Corpus (and what may be published to participants)

Training data | Training data ground truth | Test data | Test data ground truth

Software (and what may be submitted by participants)

Software source | Software executable | Software run

Submission type

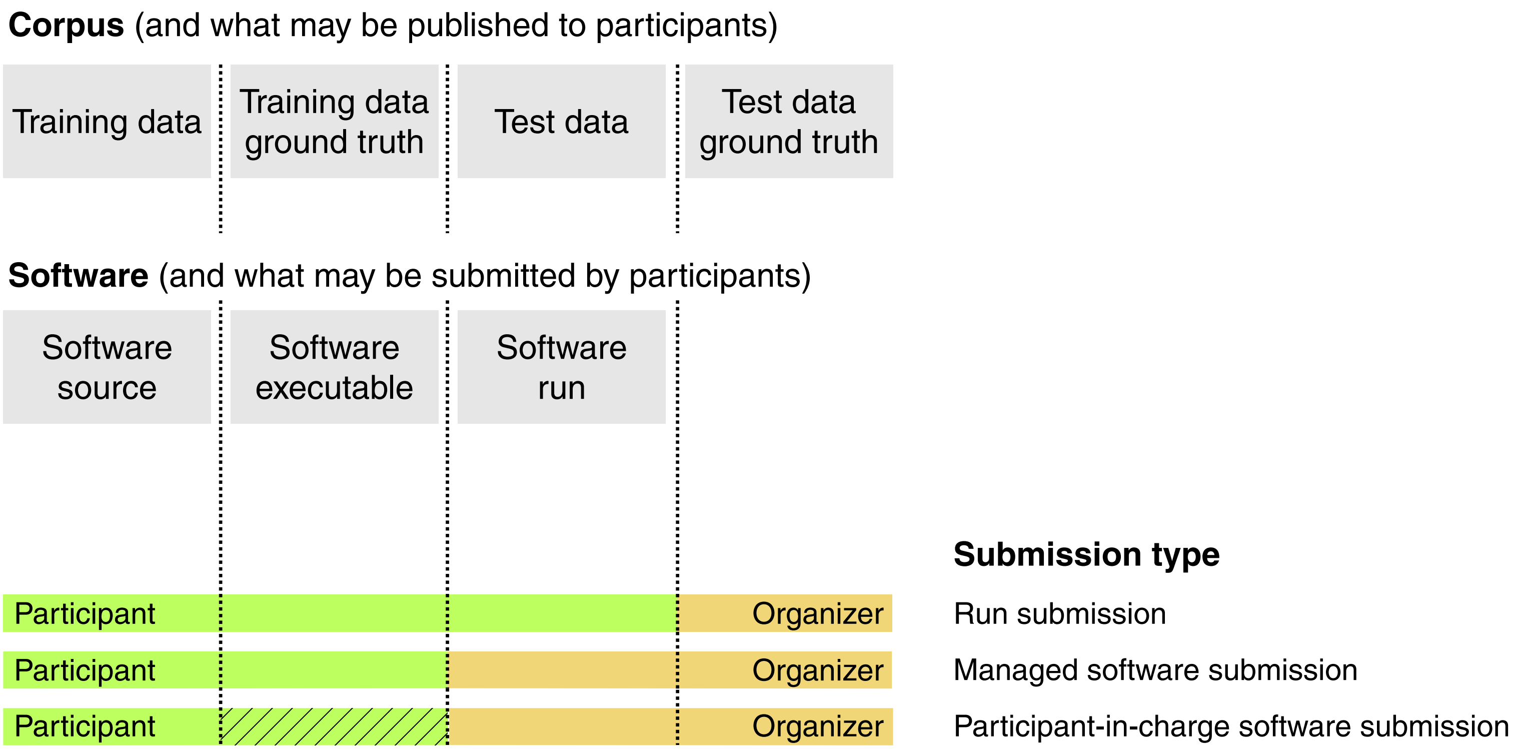
Participant — Organizer: Run submission

Participant — Organizer: Managed software submission

Participant — Organizer: Participant-in-charge software submission

About Shared Tasks

Shared Tasks by Submission Type



Corpus (and what may be published to participants)

Training data | Training data ground truth | Test data | Test data ground truth

Software (and what may be submitted by participants)

Software source | Software executable | Software run

Submission type

Participant … Organizer — Run submission

Participant … Organizer — Managed software submission

Participant … Organizer — Participant-in-charge software submission

- PAN 2009-2011 run submission
- PAN 2012 managed software submission (1 task)
- PAN 2013 managed software submission (all tasks)
- PAN 2014 participant-in-charge software submssion via TIRA

The TIRA experiment platform

The TIRA experiment platform

Software Submission Challenges ➜ Approaches

1. Environment diversity ➜ virtualization
   Support a wide variety of programming languages and operating systems.
2. Executing untrusted software ➜ virtualization
   Better be safe than sorry when executing binaries from a third party.
3. Data leakage ➜ sandboxing
   Prevent data leaking by running software in a secured environment.

The TIRA experiment platform

Software Submission Challenges ➜ Approaches

1. Environment diversity ➜ virtualization
   Support a wide variety of programming languages and operating systems.
2. Executing untrusted software ➜ virtualization
   Better be safe than sorry when executing binaries from a third party.
3. Data leakage ➜ sandboxing
   Prevent data leaking by running software in a secured environment.

4. Error handling ➜ user interface, unit testing
   Give participants the tools to find and fix their software’s errors.
5. Responsibility ➜ user interface
   Put participants back in charge of their submitted software.

The TIRA experiment platform

Software Submission Challenges ➜ Approaches

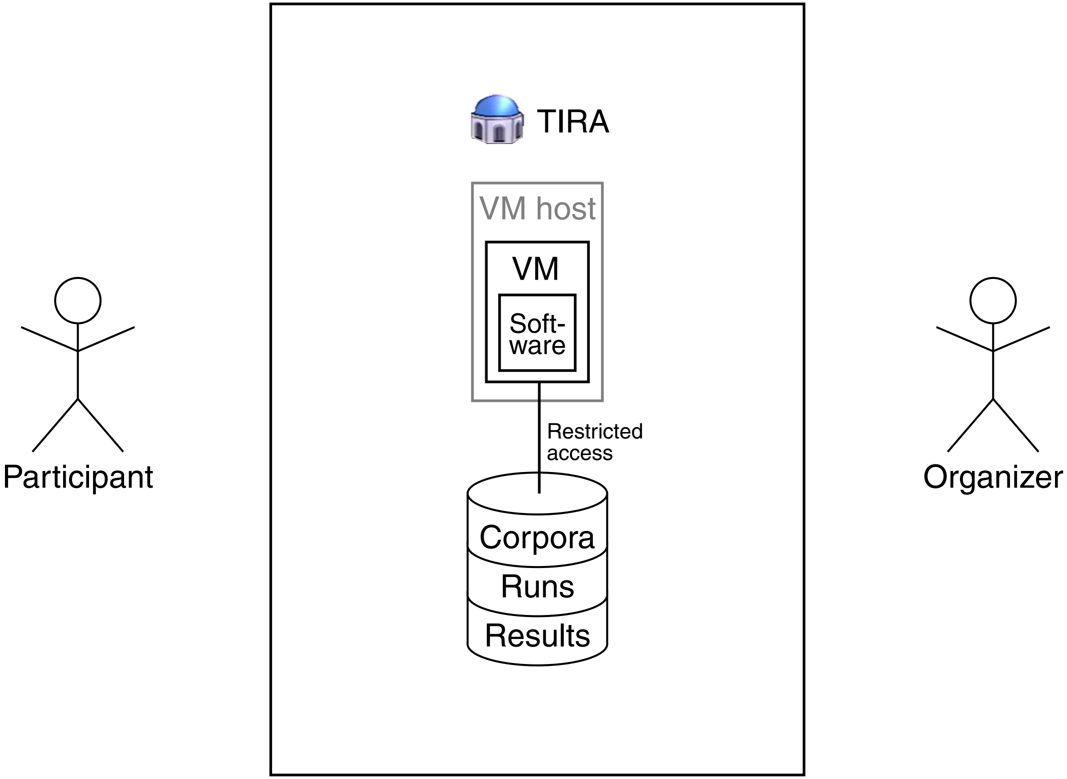
1. Environment diversity ➜ virtualization
   Support a wide variety of programming languages and operating systems.
2. Executing untrusted software ➜ virtualization
   Better be safe than sorry when executing binaries from a third party.
3. Data leakage ➜ sandboxing
   Prevent data leaking by running software in a secured environment.

4. Error handling ➜ user interface, unit testing
   Give participants the tools to find and fix their software's errors.
5. Responsibility ➜ user interface
   Put participants back in charge of their submitted software.

6. Execution cost ➜ provide hardware or raise usage fees
   We provide servers to host virtual machines.

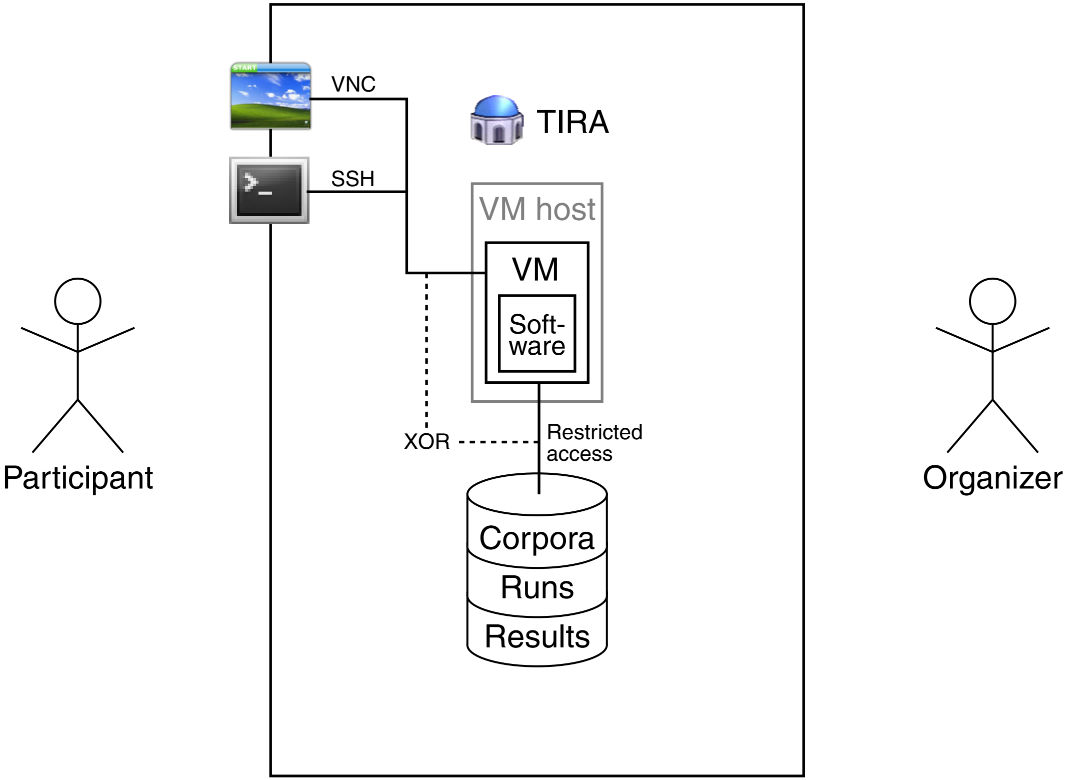
The TIRA experiment platform

System Architecture: User Interfaces



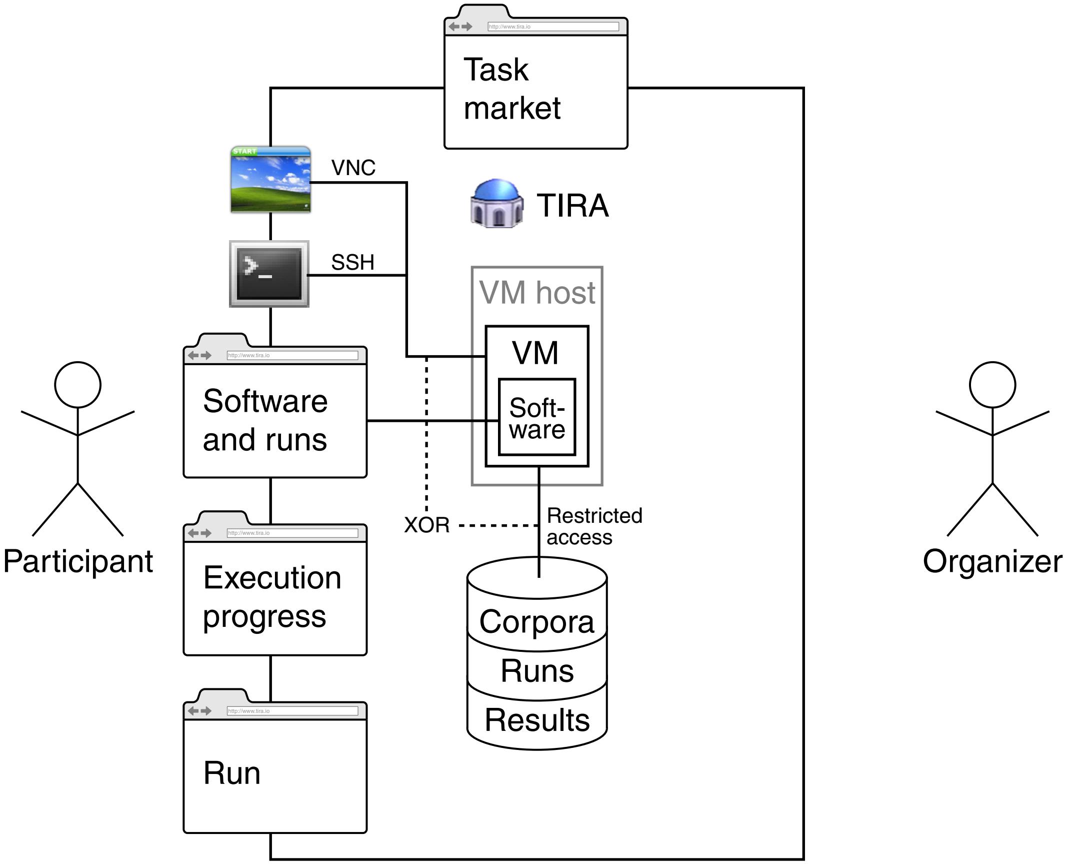
The TIRA experiment platform

System Architecture: User Interfaces



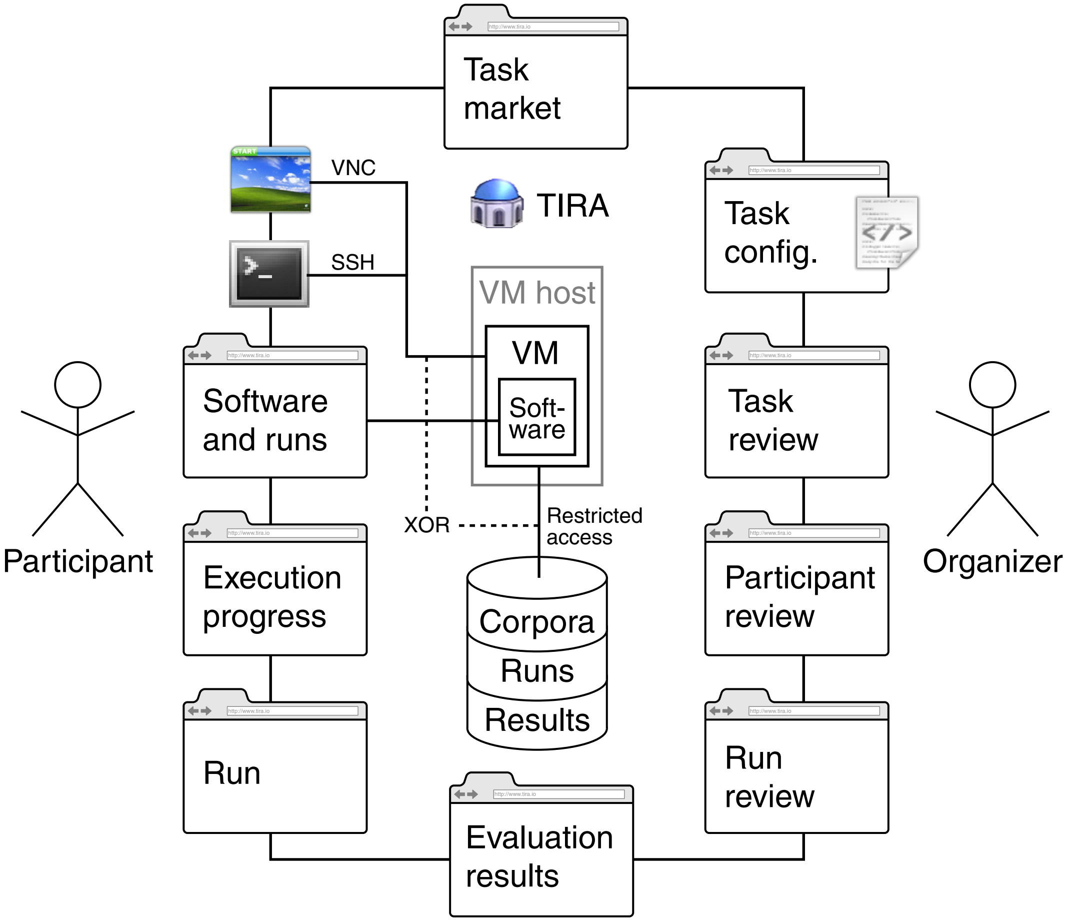
The TIRA experiment platform

System Architecture: User Interfaces



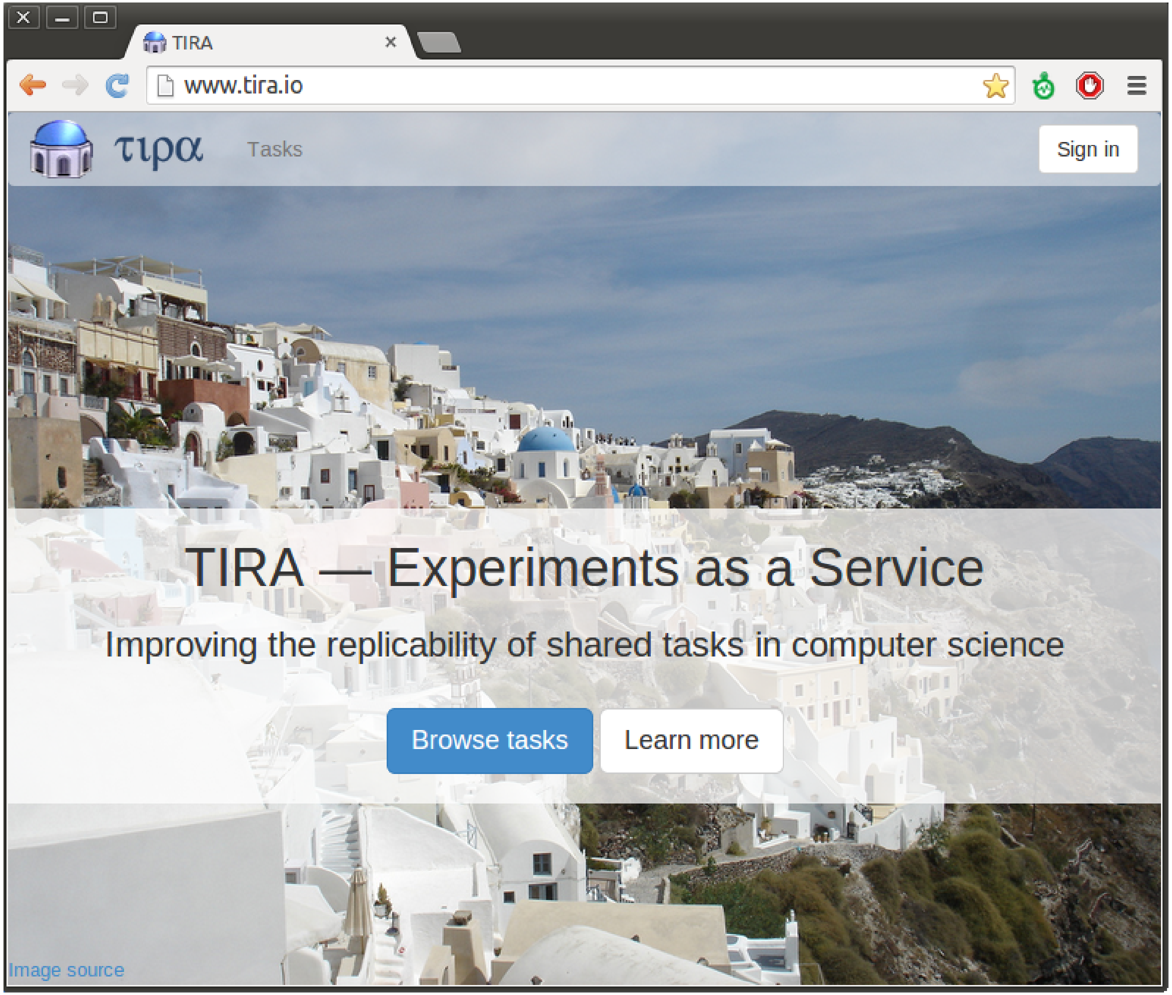
The TIRA experiment platform

System Architecture: User Interfaces



The TIRA experiment platform

Demo



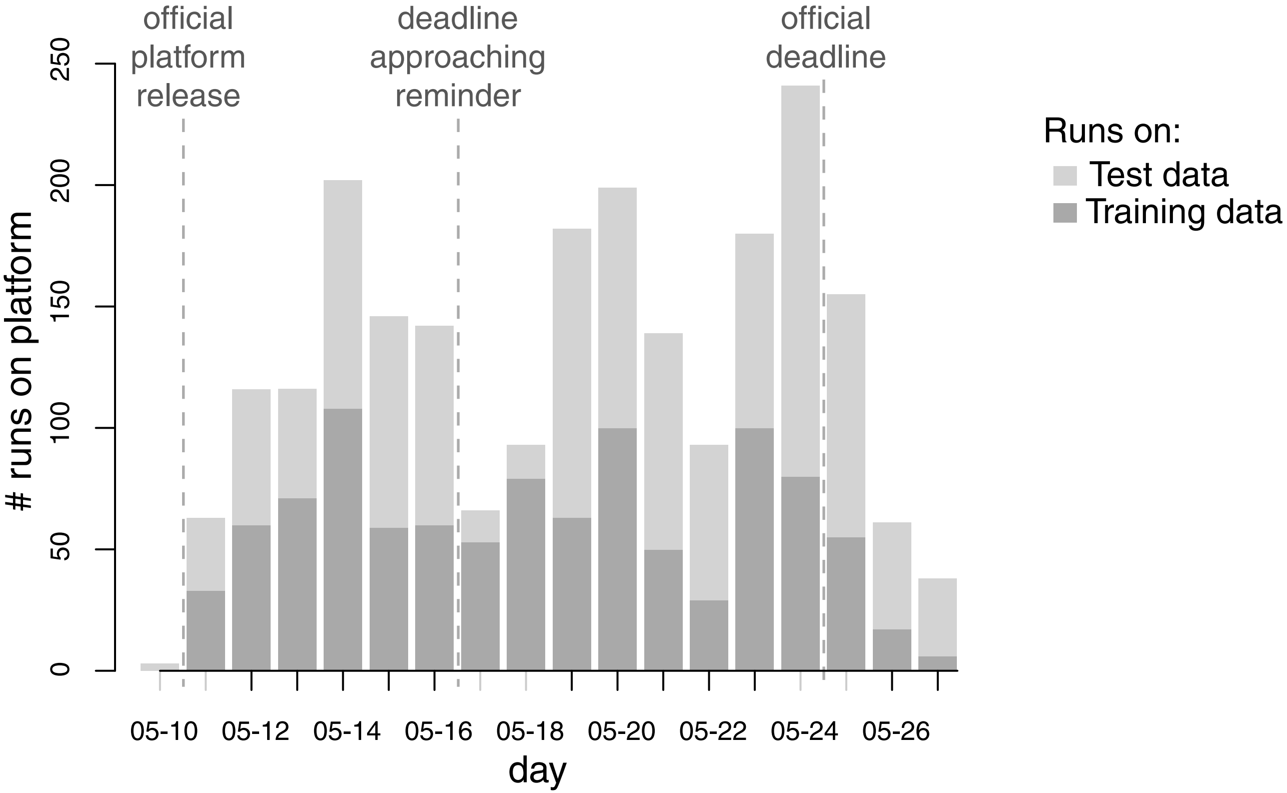
TIRA — Experiments as a Service

Improving the replicability of shared tasks in computer science

[www.tira.io]

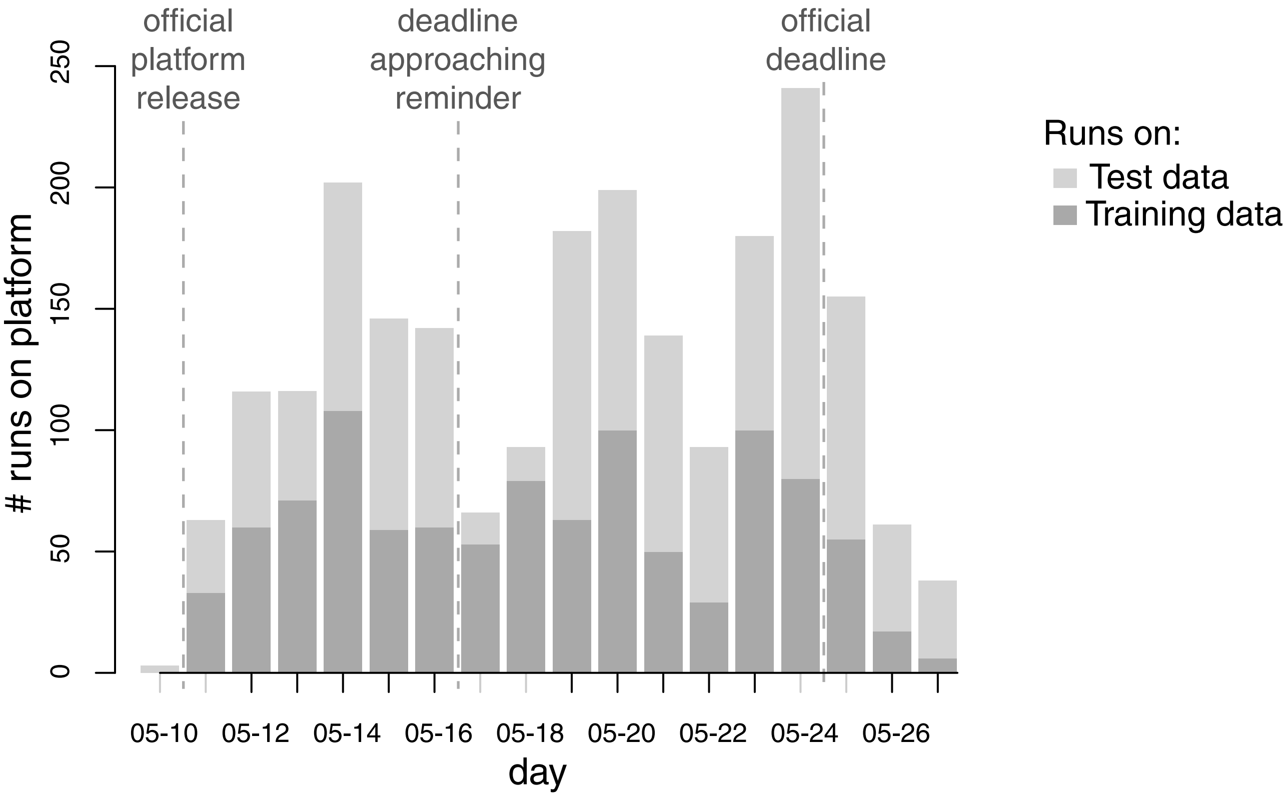
The TIRA experiment platform

Log Analysis



The TIRA experiment platform

Log Analysis



New success indicator for shared tasks

- participant engagement (real-time, personalized)

Summary

The PAN Competition

PAN is a network around digital text forensics.

Mission

- Foster research and development in our tasks
- Push the limits of evaluating them
- Improve methodology for lab-style evaluations

Tasks

- Author Profiling (Given a document, what are its author's demographics?)
- Author Identification (Given a document, who wrote it?)
- Plagiarism Detection (Given a document, is it an original?)

Summary

The PAN Competition

Statistics	**ALLC**	**SEPLN**	**FIRE**			**CLEF**				
	2004	2009	2011	2012	2013	2010	2011	2012	2013	2014
Task(s)	1	1	1	1	1	2	3	3	3	3
Follower		78				151	181	232	286	302
Registrations	11	21	6	12	16	53	52	68	110	103
Runs/Software	13	14	6	8	8	27	27	48	58	57
Notebooks	8	11	6	2	6	22	22	34	47	36
Attendees	5	18	6	30	50	25	36	61	58	

Take-away messages

- Shared tasks are understudied
- Most shared tasks invite run submissions
- Software submissions feasible at scale iff assisted by technology
- TIRA is the first platform to handle software submissions at scale

Summary

The PAN Competition

Statistics	ALLC	SEPLN	FIRE			CLEF				
	2004	2009	2011	2012	2013	2010	2011	2012	2013	2014
Task(s)	1	1	1	1	1	2	3	3	3	3
Follower		78				151	181	232	286	302
Registrations	11	21	6	12	16	53	52	68	110	103
Runs/Software	13	14	6	8	8	27	27	48	58	57
Notebooks	8	11	6	2	6	22	22	34	47	36
Attendees	5	18	6	30	50	25	36	61	58	

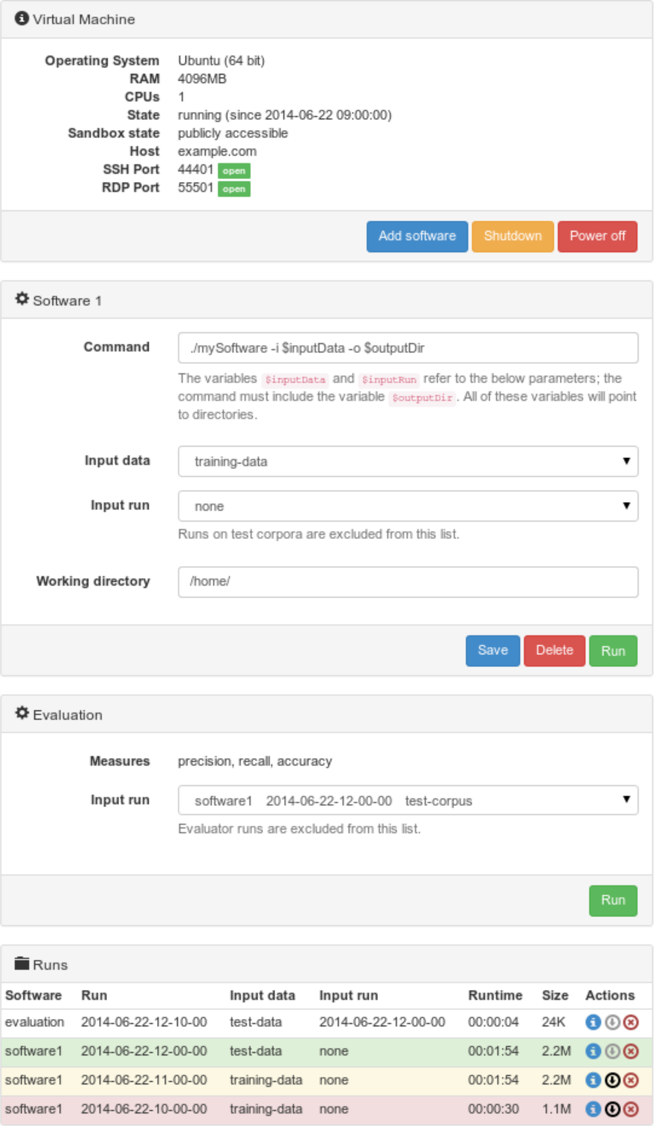
Take-away messages

- Shared tasks are understudied
- Most shared tasks invite run submissions
- Software submissions feasible at scale iff assisted by technology
- TIRA is the first platform to handle software submissions at scale

Thank you for your attention!

TIRA’s User Interfaces

Software and Runs



Virtual Machine

Operating System Ubuntu (64 bit)
RAM 4096MB
CPUs 1
State running (since 2014-06-22 09:00:00)
Sandbox state publicly accessible
Host example.com
SSH Port 44401 open
RDP Port 55501 open

Software 1

Command ./mySoftware -i $inputData -o $outputDir

The variables `$inputData` and `$inputRun` refer to the below parameters; the command must include the variable `$outputDir`. All of these variables will point to directories.

Input data training-data

Input run none

Runs on test corpora are excluded from this list.

Working directory /home/

Evaluation

Measures precision, recall, accuracy

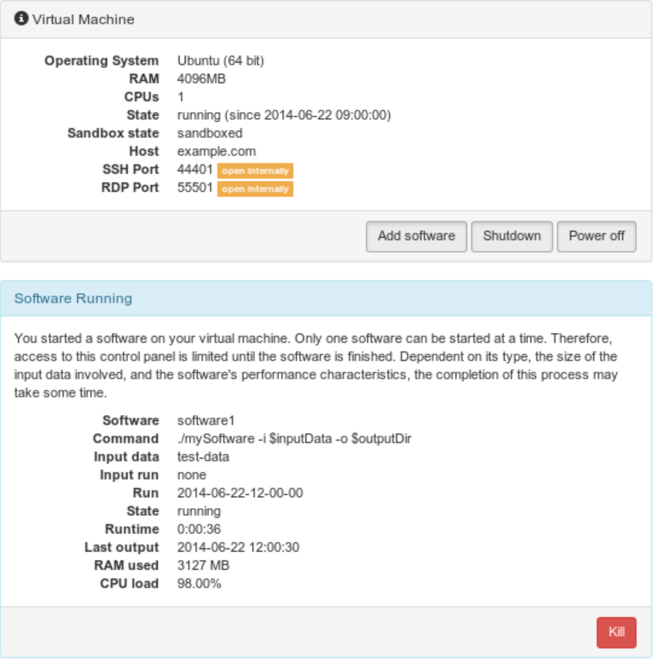
Input run software1 2014-06-22-12-00-00 test-corpus

Evaluator runs are excluded from this list.

Runs

Software	Run	Input data	Input run	Runtime	Size	Actions
evaluation	2014-06-22-12-10-00	test-data	2014-06-22-12-00-00	00:00:04	24K	ⓘ ⬇ ⊗
software1	2014-06-22-12-00-00	test-data	none	00:01:54	2.2M	ⓘ ⬇ ⊗
software1	2014-06-22-11-00-00	training-data	none	00:01:54	2.2M	ⓘ ⬇ ⊗
software1	2014-06-22-10-00-00	training-data	none	00:00:30	1.1M	ⓘ ⬇ ⊗

Execution Progress



Virtual Machine

Operating System Ubuntu (64 bit)
RAM 4096MB
CPUs 1
State running (since 2014-06-22 09:00:00)
Sandbox state sandboxed
Host example.com
SSH Port 44401 open internally
RDP Port 55501 open internally

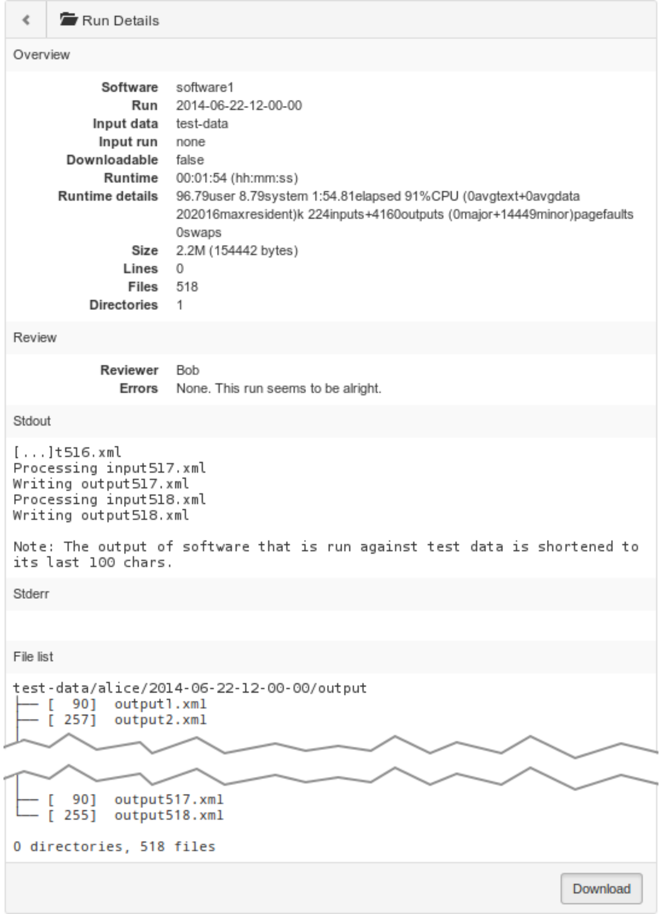
Software Running

You started a software on your virtual machine. Only one software can be started at a time. Therefore, access to this control panel is limited until the software is finished. Dependent on its type, the size of the input data involved, and the software's performance characteristics, the completion of this process may take some time.

Software software1
Command ./mySoftware -i $inputData -o $outputDir
Input data test-data
Input run none
Run 2014-06-22-12-00-00
State running
Runtime 0:00:36
Last output 2014-06-22 12:00:30
RAM used 3127 MB
CPU load 98.00%

TIRA’s User Interfaces

Run



Overview

Software	software1
Run	2014-06-22-12-00-00
Input data	test-data
Input run	none
Downloadable	false
Runtime	00:01:54 (hh:mm:ss)
Runtime details	96.79user 8.79system 1:54.81elapsed 91%CPU (0avgtext+0avgdata 202016maxresident)k 224inputs+4160outputs (0major+14449minor)pagefaults 0swaps
Size	2.2M (154442 bytes)
Lines	0
Files	518
Directories	1

Review

Reviewer	Bob
Errors	None. This run seems to be alright.

Stdout

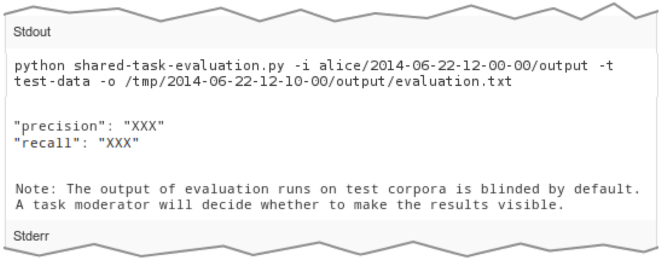
```
[...]t516.xml
Processing input517.xml
Writing output517.xml
Processing input518.xml
Writing output518.xml

Note: The output of software that is run against test data is shortened to
its last 100 chars.
```

Stderr

File list

```
test-data/alice/2014-06-22-12-00-00/output
├── [  90]  output1.xml
├── [ 257]  output2.xml

├── [  90]  output517.xml
└── [ 255]  output518.xml

0 directories, 518 files
```

Evaluation Run (excerpt)



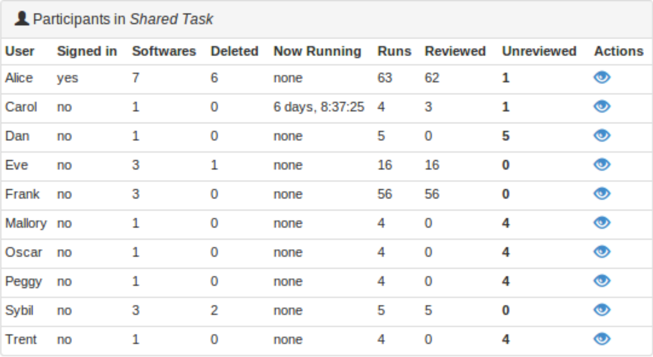
Stdout

```
python shared-task-evaluation.py -i alice/2014-06-22-12-00-00/output -t
test-data -o /tmp/2014-06-22-12-10-00/output/evaluation.txt


"precision": "XXX"
"recall": "XXX"


Note: The output of evaluation runs on test corpora is blinded by default.
A task moderator will decide whether to make the results visible.
```

Stderr

TIRA’s User Interfaces

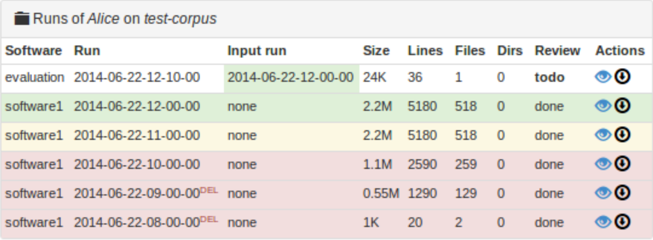
Task Review



Participants in *Shared Task*

User	Signed in	Softwares	Deleted	Now Running	Runs	Reviewed	Unreviewed	Actions
Alice	yes	7	6	none	63	62	**1**	👁
Carol	no	1	0	6 days, 8:37:25	4	3	**1**	👁
Dan	no	1	0	none	5	0	**5**	👁
Eve	no	3	1	none	16	16	**0**	👁
Frank	no	3	0	none	56	56	**0**	👁
Mallory	no	1	0	none	4	0	**4**	👁
Oscar	no	1	0	none	4	0	**4**	👁
Peggy	no	1	0	none	4	0	**4**	👁
Sybil	no	3	2	none	5	5	**0**	👁
Trent	no	1	0	none	4	0	**4**	👁

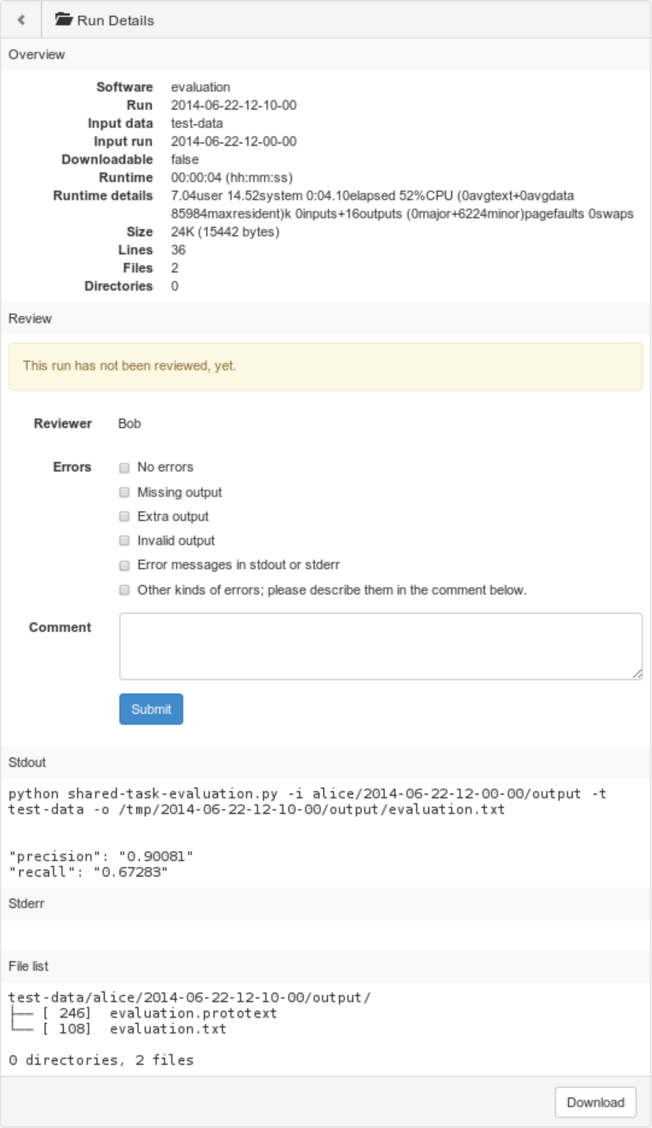
Particpant Review



Runs of *Alice* on *test-corpus*

Software	Run	Input run	Size	Lines	Files	Dirs	Review	Actions
evaluation	2014-06-22-12-10-00	2014-06-22-12-00-00	24K	36	1	0	**todo**	👁 ⬇
software1	2014-06-22-12-00-00	none	2.2M	5180	518	0	done	👁 ⬇
software1	2014-06-22-11-00-00	none	2.2M	5180	518	0	done	👁 ⬇
software1	2014-06-22-10-00-00	none	1.1M	2590	259	0	done	👁 ⬇
software1	2014-06-22-09-00-00$^{\text{DEL}}$	none	0.55M	1290	129	0	done	👁 ⬇
software1	2014-06-22-08-00-00$^{\text{DEL}}$	none	1K	20	2	0	done	👁 ⬇

Run Review



Overview

Software	evaluation
Run	2014-06-22-12-10-00
Input data	test-data
Input run	2014-06-22-12-00-00
Downloadable	false
Runtime	00:00:04 (hh:mm:ss)
Runtime details	7.04user 14.52system 0:04.10elapsed 52%CPU (0avgtext+0avgdata 85984maxresident)k 0inputs+16outputs (0major+6224minor)pagefaults 0swaps
Size	24K (15442 bytes)
Lines	36
Files	2
Directories	0

Review

This run has not been reviewed, yet.

Reviewer Bob

Errors
- □ No errors
- □ Missing output
- □ Extra output
- □ Invalid output
- □ Error messages in stdout or stderr
- □ Other kinds of errors; please describe them in the comment below.

Comment

Stdout

```
python shared-task-evaluation.py -i alice/2014-06-22-12-00-00/output -t
test-data -o /tmp/2014-06-22-12-10-00/output/evaluation.txt


"precision": "0.90081"
"recall": "0.67283"
```

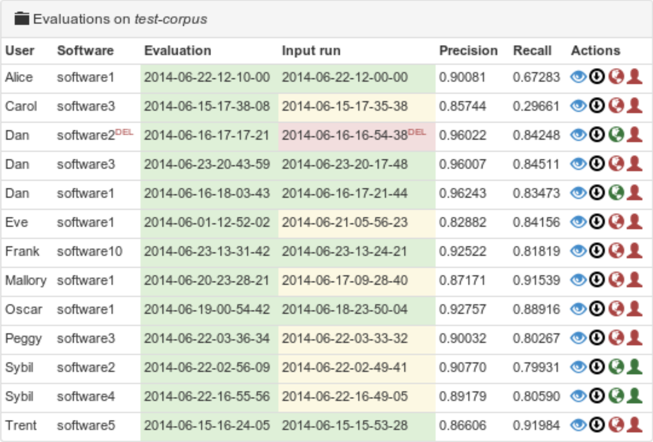
Stderr

File list

```
test-data/alice/2014-06-22-12-10-00/output/
├── [ 246]  evaluation.prototext
└── [ 108]  evaluation.txt

0 directories, 2 files
```

TIRA’s User Interfaces

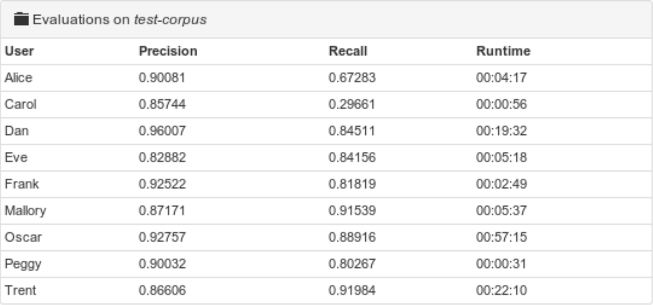
Evaluation Results Review



Evaluations on *test-corpus*

User	Software	Evaluation	Input run	Precision	Recall	Actions
Alice	software1	2014-06-22-12-10-00	2014-06-22-12-00-00	0.90081	0.67283	👁 ⬇ 🔴 🟥
Carol	software3	2014-06-15-17-38-08	2014-06-15-17-35-38	0.85744	0.29661	👁 ⬇ 🔴 🟥
Dan	software2$^{\text{DEL}}$	2014-06-16-17-17-21	2014-06-16-16-54-38$^{\text{DEL}}$	0.96022	0.84248	👁 ⬇ 🟢 🟥
Dan	software3	2014-06-23-20-43-59	2014-06-23-20-17-48	0.96007	0.84511	👁 ⬇ 🔴 🟥
Dan	software1	2014-06-16-18-03-43	2014-06-16-17-21-44	0.96243	0.83473	👁 ⬇ 🟢 🟥
Eve	software1	2014-06-01-12-52-02	2014-06-21-05-56-23	0.82882	0.84156	👁 ⬇ 🔴 🟥
Frank	software10	2014-06-23-13-31-42	2014-06-23-13-24-21	0.92522	0.81819	👁 ⬇ 🔴 🟥
Mallory	software1	2014-06-20-23-28-21	2014-06-17-09-28-40	0.87171	0.91539	👁 ⬇ 🔴 🟥
Oscar	software1	2014-06-19-00-54-42	2014-06-18-23-50-04	0.92757	0.88916	👁 ⬇ 🔴 🟥
Peggy	software3	2014-06-22-03-36-34	2014-06-22-03-33-32	0.90032	0.80267	👁 ⬇ 🔴 🟥
Sybil	software2	2014-06-22-02-56-09	2014-06-22-02-49-41	0.90770	0.79931	👁 ⬇ 🟢 🟩
Sybil	software4	2014-06-22-16-55-56	2014-06-22-16-49-05	0.89179	0.80590	👁 ⬇ 🟢 🟩
Trent	software5	2014-06-15-16-24-05	2014-06-15-15-53-28	0.86606	0.91984	👁 ⬇ 🔴 🟥

Evaluation Results (published)



Evaluations on *test-corpus*

User	Precision	Recall	Runtime
Alice	0.90081	0.67283	00:04:17
Carol	0.85744	0.29661	00:00:56
Dan	0.96007	0.84511	00:19:32
Eve	0.82882	0.84156	00:05:18
Frank	0.92522	0.81819	00:02:49
Mallory	0.87171	0.91539	00:05:37
Oscar	0.92757	0.88916	00:57:15
Peggy	0.90032	0.80267	00:00:31
Trent	0.86606	0.91984	00:22:10