TIRA: Configuring, Executing, and Disseminating Information Retrieval Experiments

Tim Gollub
Benno Stein
Steven Burrows
Dennis Hoppe

Webis Group www.webis.de
Bauhaus-Universität Weimar

TIRA: Configuring, Executing, and Disseminating Information Retrieval Experiments

Introduction

Quotes

❑ A longitudinal study has shown consistent selection of weak baselines in ad-hoc retrieval tasks leading to "**improvements that don't add up**".

[Armstrong et al., 2009]

❑ A polarizing article describes how biases in research approaches lead to the consideration of "**why most published research findings are false**".

[Ioannidis, 2005]

❑ The SWIRL 2002 meeting of 45 information retrieval researchers considered evaluation as a "**perennial issue in information retrieval**" and that there is a clear need for a "**community evaluation service**".

[Allan et al., 2012]

❑ "We have to **explore systematically** the **independent parameters** of experiments."

[Fuhr, Salton Award Speech, SIGIR 2012]

Introduction

Quotes

- A longitudinal study has shown consistent selection of weak baselines in ad-hoc retrieval tasks leading to "**improvements that don't add up**".

  [Armstrong et al., 2009]

- A polarizing article describes how biases in research approaches lead to the consideration of "**why most published research findings are false**".

  [Ioannidis, 2005]

- The SWIRL 2002 meeting of 45 information retrieval researchers considered evaluation as a "**perennial issue in information retrieval**" and that there is a clear need for a "**community evaluation service**".

  [Allan et al., 2012]

- "We have to **explore systematically** the **independent parameters** of experiments."

  [Fuhr, Salton Award Speech, SIGIR 2012]

Introduction

❑ A longitudinal study has shown consistent selection of weak baselines in ad-hoc retrieval tasks leading to "**improvements that don't add up**".

[Armstrong et al., 2009]

❑ A polarizing article describes how biases in research approaches lead to the consideration of "**why most published research findings are false**".

[Ioannidis, 2005]

❑ The SWIRL 2002 meeting of 45 information retrieval researchers considered evaluation as a "**perennial issue in information retrieval**" and that there is a clear need for a "**community evaluation service**".

[Allan et al., 2012]

❑ "We have to **explore systematically** the **independent parameters** of experiments."

[Fuhr, Salton Award Speech, SIGIR 2012]

Introduction

- A longitudinal study has shown consistent selection of weak baselines in ad-hoc retrieval tasks leading to "**improvements that don't add up**".

    [Armstrong et al., 2009]

- A polarizing article describes how biases in research approaches lead to the consideration of "**why most published research findings are false**".

    [Ioannidis, 2005]

- The SWIRL 2002 meeting of 45 information retrieval researchers considered evaluation as a "**perennial issue in information retrieval**" and that there is a clear need for a "**community evaluation service**".

    [Allan et al., 2012]

- "We have to **explore systematically** the **independent parameters** of experiments."
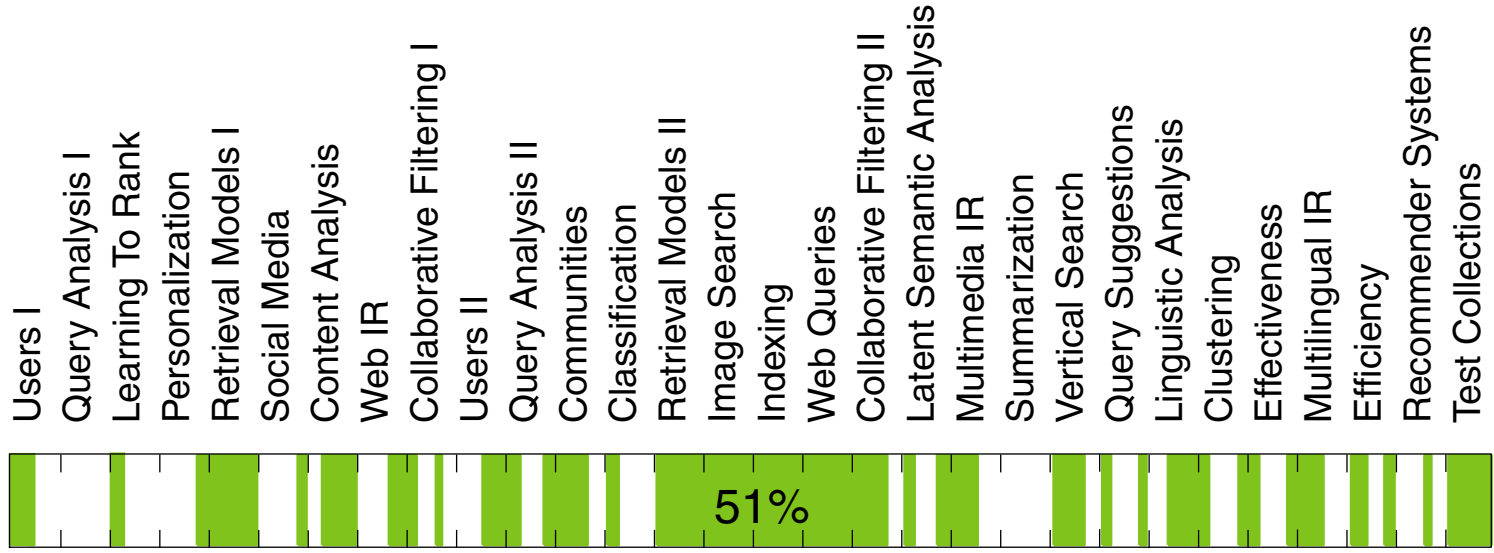
    [Fuhr, Salton Award Speech, SIGIR 2012]

Introduction

Survey of 108 Full Papers at SIGIR 2011

Users I | Query Analysis I | Learning To Rank | Personalization | Retrieval Models I | Social Media | Content Analysis | Web IR | Collaborative Filtering I | Users II | Query Analysis II | Communities | Classification | Retrieval Models II | Image Search | Indexing | Web Queries | Collaborative Filtering II | Latent Semantic Analysis | Multimedia IR | Summarization | Vertical Search | Query Suggestions | Linguistic Analysis | Clustering | Effectiveness | Multilingual IR | Efficiency | Recommender Systems | Test Collections

Introduction

Survey of 108 Full Papers at SIGIR 2011

Users I
Query Analysis I
Learning To Rank
Personalization
Retrieval Models I
Social Media
Content Analysis
Web IR
Collaborative Filtering I
Users II
Query Analysis II
Communities
Classification
Retrieval Models II
Image Search
Indexing
Web Queries
Collaborative Filtering II
Latent Semantic Analysis
Multimedia IR
Summarization
Vertical Search
Query Suggestions
Linguistic Analysis
Clustering
Effectiveness
Multilingual IR
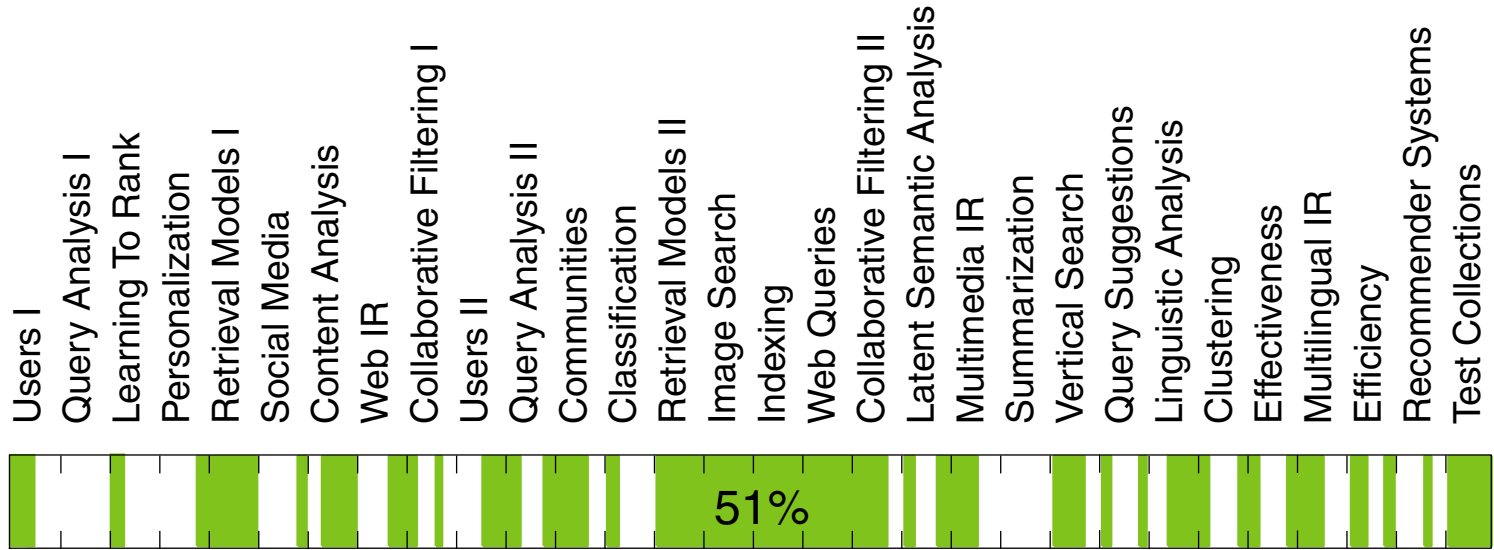Efficiency
Recommender Systems
Test Collections
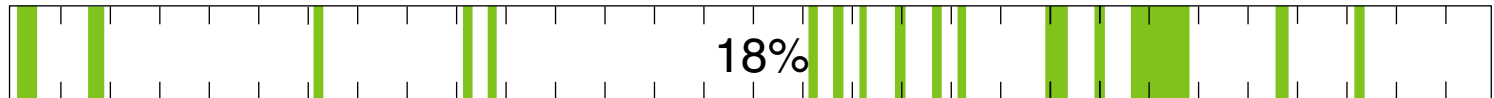
51%

Provision of experiment data

Introduction

Survey of 108 Full Papers at SIGIR 2011
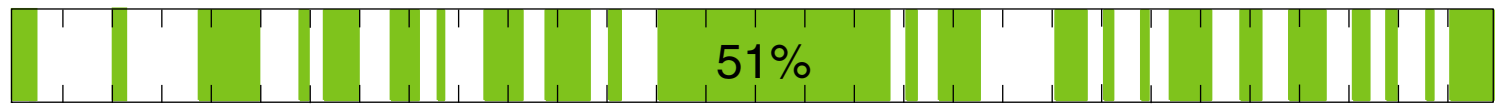

Provision of experiment data — 51%
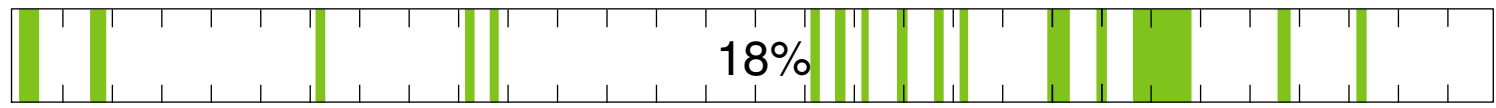
Provision of experiment software — 18%
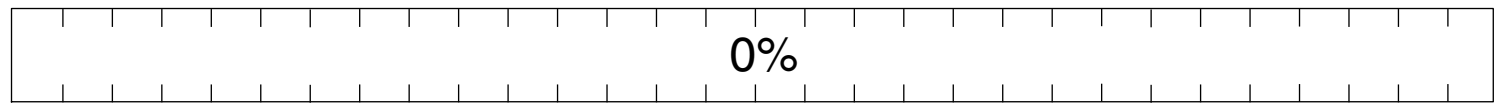
Introduction

Survey of 108 Full Papers at SIGIR 2011

Users I · Query Analysis I · Learning To Rank · Personalization · Retrieval Models I · Social Media · Content Analysis · Web IR · Collaborative Filtering I · Users II · Query Analysis II · Communities · Classification · Retrieval Models II · Image Search · Indexing · Web Queries · Collaborative Filtering II · Latent Semantic Analysis · Multimedia IR · Summarization · Vertical Search · Query Suggestions · Linguistic Analysis · Clustering · Effectiveness · Multilingual IR · Efficiency · Recommender Systems · Test Collections

51%

Provision of experiment data
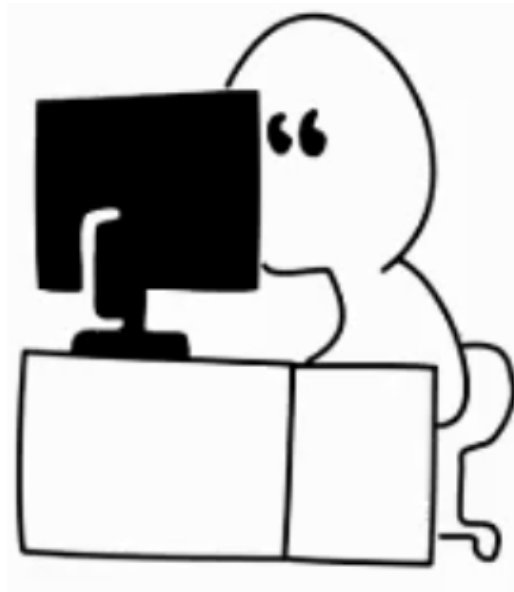
18%

Provision of experiment software

0%

Provision of experiment service

Introduction

Incentives for Reproducible Research

❑ Increase acknowledgment for publishing experiments, data, and software.

   – Encourage a paradigm shift towards open science.

❑ Decrease the overhead of publishing experiments.

   – The concept of TIRA is to provide "experiments as a service".

Architecture

Design Goals

1. Local Instantiation

- ❑ Enables public research on private data.
- ❑ Enables comparisons with private software.

2. Unique Resource Identifiers

- ❑ Enables linkage of experimental results in papers with the respective experiment service.
- ❑ Enables reproduction of results on the basis of the resource identifier (digital preservation).
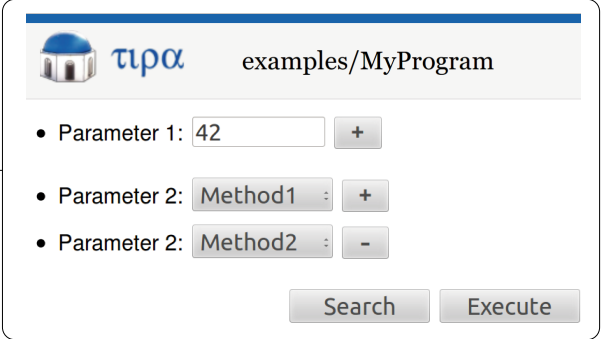
3. Multivalued Configuration

- ❑ Enables the specification of whole experiment series.

(1)

localhost:2306/programs/examples/MyProgram?p1=42&p2=Method1&p2=Method2

(2)

(3)

τιρα   examples/MyProgram

- Parameter 1: `42`   `+`
- Parameter 2: `Method1` ⁝ `+`
- Parameter 2: `Method2` ⁝ `-`

`Search`   `Execute`

(4)   `tira@node1:~$ ./myprogram.sh -p1 42 -p2 "method1"`

(5)   `tira@node2:~$ ./myprogram.sh -p1 42 -p2 "method2"`

(6)

Parameter 1	Parameter 2	Output Directory	Performance
42	Method1	output-directory	0.89
42	Method2	output-directory	0.71

Architecture

Design Goals

1. Local Instantiation

- ❑ Enables public research on private data.
- ❑ Enables comparisons with private software.

2. Unique Resource Identifiers

- ❑ Enables linkage of experimental results in papers with the respective experiment service.
- ❑ Enables reproduction of results on the basis of the resource identifier (digital preservation).
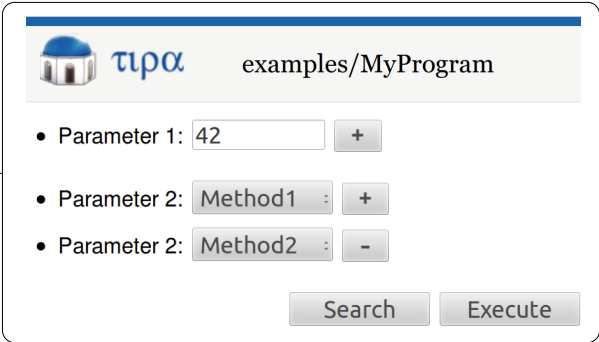
3. Multivalued Configuration

- ❑ Enables the specification of whole experiment series.

1

localhost:2306/programs/examples/MyProgram?p1=42&p2=Method1&p2=Method2

2

3

τιρα   examples/MyProgram

- Parameter 1: 42   +
- Parameter 2: Method1 ⫶ +
- Parameter 2: Method2 ⫶ −

Search   Execute

4
```
tira@node1:~$ ./myprogram.sh -p1 42 -p2 "method1"
```

5
```
tira@node2:~$ ./myprogram.sh -p1 42 -p2 "method2"
```

6

Parameter 1	Parameter 2	Output Directory	Performance
42	Method1	output-directory	0.89
42	Method2	output-directory	0.71

Architecture

Design Goals

1. Local Instantiation

- ❏ Enables public research on private data.
- ❏ Enables comparisons with private software.

2. Unique Resource Identifiers

- ❏ Enables linkage of experimental results in papers with the respective experiment service.
- ❏ Enables reproduction of results on the basis of the resource identifier (digital preservation).
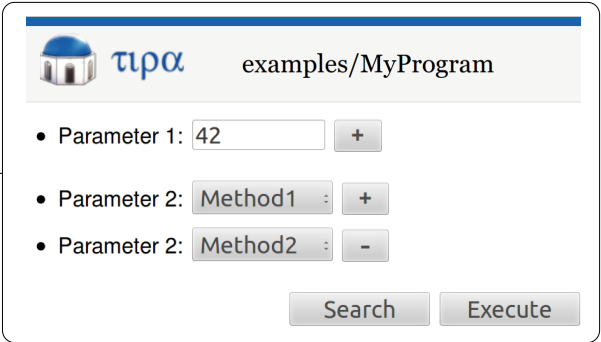
3. Multivalued Configuration

- ❏ Enables the specification of whole experiment series.

①
②

`localhost:2306/programs/examples/MyProgram?p1=42&p2=Method1&p2=Method2`

③

τιρα examples/MyProgram

- Parameter 1: `42` [+]
- Parameter 2: `Method1` ⋮ [+]
- Parameter 2: `Method2` ⋮ [-]

[Search] [Execute]

④ `tira@node1:~$ ./myprogram.sh -p1 42 -p2 "method1"`

⑤ `tira@node2:~$ ./myprogram.sh -p1 42 -p2 "method2"`

⑥

Parameter 1	Parameter 2	Output Directory	Performance
42	Method1	output-directory	0.89
42	Method2	output-directory	0.71

Architecture

Design Goals (continued)

4. System Independence

- ❏ Enables a widespread usage of the platform.
- ❏ Enables the deployment of any experiment software without internal modifications.

5. Distributed Execution

- ❏ Enables efficient computation of pending experiments.

6. Result Storage

- ❏ Enables retrieval and maintenance of raw experiment results.

… and **Peer to Peer Collaboration**

- ❏ Conduct shared work on the same platform.

1

2

`localhost:2306/programs/examples/MyProgram?p1=42&p2=Method1&p2=Method2`

3

τιρα   examples/MyProgram

- Parameter 1: 42   +
- Parameter 2: Method1 :   +
- Parameter 2: Method2 :   -

Search   Execute

4   `tira@node1:~$ ./myprogram.sh -p1 42 -p2 "method1"`

5   `tira@node2:~$ ./myprogram.sh -p1 42 -p2 "method2"`

6

Parameter 1	Parameter 2	Output Directory	Performance
42	Method1	output-directory	0.89
42	Method2	output-directory	0.71

Architecture

Design Goals (continued)

4. System Independence

- ❑ Enables a widespread usage of the platform.
- ❑ Enables the deployment of any experiment software without internal modifications.
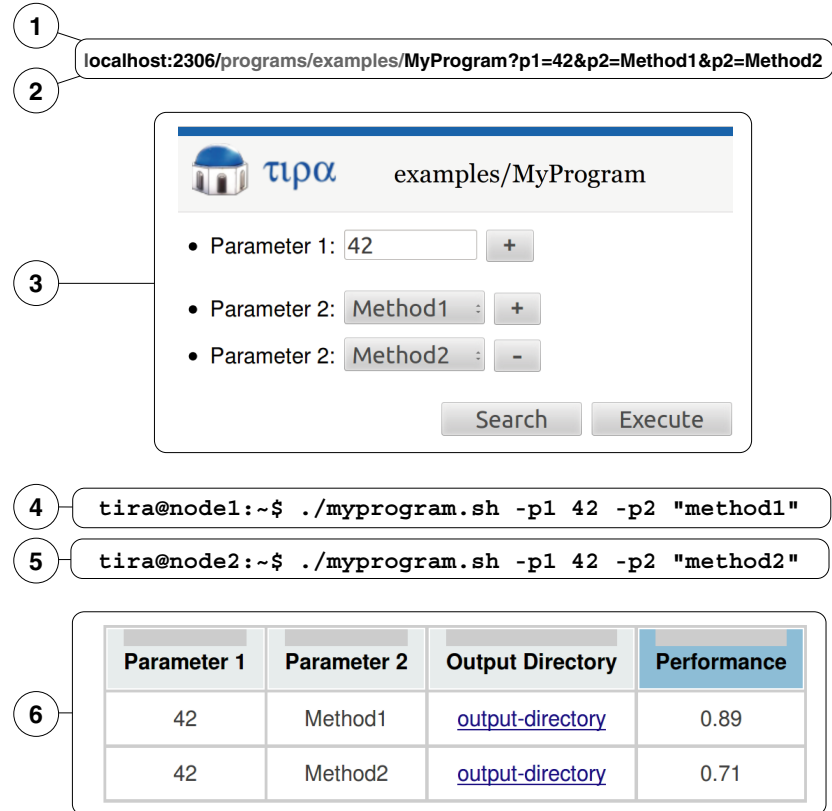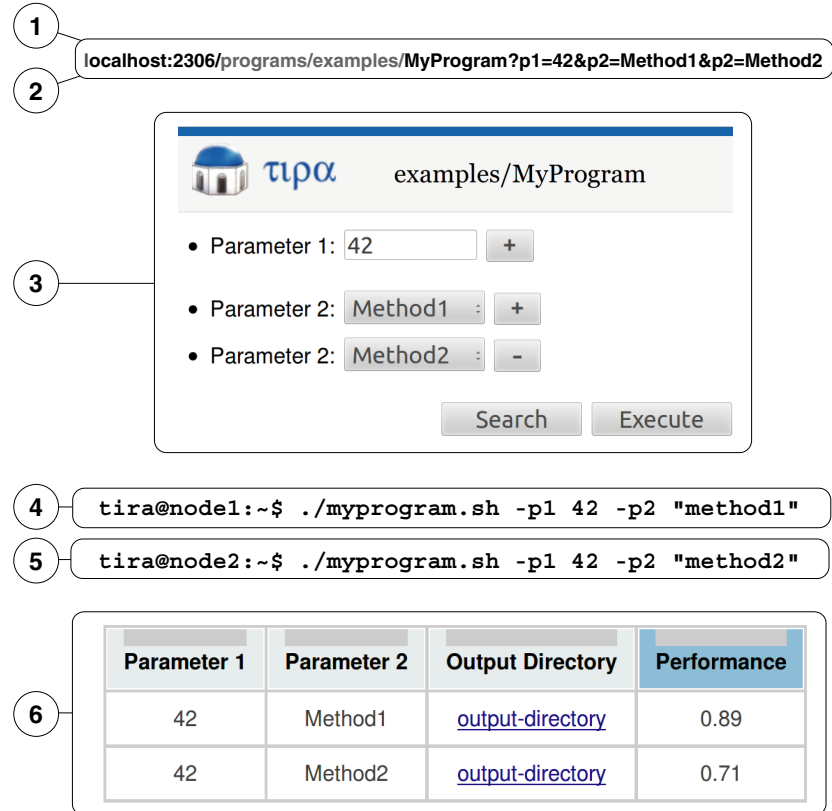
5. Distributed Execution

- ❑ Enables efficient computation of pending experiments.

6. Result Storage

- ❑ Enables retrieval and maintenance of raw experiment results.

… and Peer to Peer Collaboration

- ❑ Conduct shared work on the same platform.

1 **2**

`localhost:2306/programs/examples/MyProgram?p1=42&p2=Method1&p2=Method2`

3

τιρα   examples/MyProgram

- Parameter 1: `42`   `+`
- Parameter 2: `Method1`   `+`
- Parameter 2: `Method2`   `-`

`Search`   `Execute`

4
```
tira@node1:~$ ./myprogram.sh -p1 42 -p2 "method1"
```

5
```
tira@node2:~$ ./myprogram.sh -p1 42 -p2 "method2"
```

6

Parameter 1	Parameter 2	Output Directory	Performance
42	Method1	output-directory	0.89
42	Method2	output-directory	0.71

Architecture

Design Goals (continued)

4. System Independence

- ❑ Enables a widespread usage of the platform.
- ❑ Enables the deployment of any experiment software without internal modifications.

5. Distributed Execution

- ❑ Enables efficient computation of pending experiments.

6. Result Storage

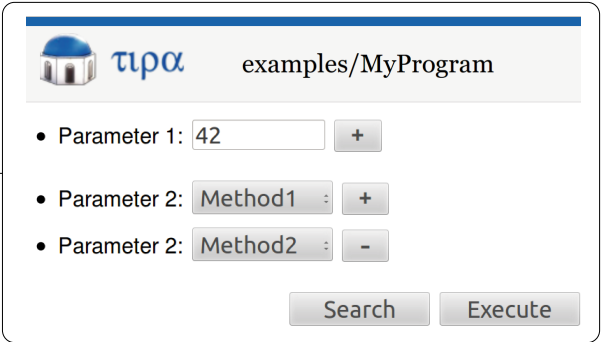- ❑ Enables retrieval and maintenance of raw experiment results.

. . . and **Peer to Peer Collaboration**

- ❑ Conduct shared work on the same platform.

1

2

`localhost:2306/programs/examples/MyProgram?p1=42&p2=Method1&p2=Method2`

3

τιρα   examples/MyProgram

- Parameter 1: 42   +
- Parameter 2: Method1 ⋮ +
- Parameter 2: Method2 ⋮ −

Search   Execute

4

```
tira@node1:~$ ./myprogram.sh -p1 42 -p2 "method1"
```

5

```
tira@node2:~$ ./myprogram.sh -p1 42 -p2 "method2"
```

6

Parameter 1	Parameter 2	Output Directory	Performance
42	Method1	output-directory	0.89
42	Method2	output-directory	0.71

Architecture

Design Goals (continued)

4. System Independence

- ❏ Enables a widespread usage of the platform.
- ❏ Enables the deployment of any experiment software without internal modifications.

5. Distributed Execution

- ❏ Enables efficient computation of pending experiments.

6. Result Storage

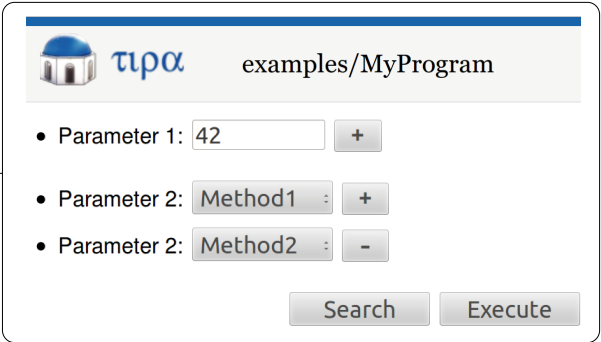- ❏ Enables retrieval and maintenance of raw experiment results.

. . . and **Peer to Peer Collaboration**

- ❏ Conduct shared work on the same platform.

1

2

```
localhost:2306/programs/examples/MyProgram?p1=42&p2=Method1&p2=Method2
```

3

τιρα   examples/MyProgram

- Parameter 1: [42]   [+]
- Parameter 2: [Method1 ⫶]   [+]
- Parameter 2: [Method2 ⫶]   [-]

[Search]   [Execute]

4
```
tira@node1:~$ ./myprogram.sh -p1 42 -p2 "method1"
```

5
```
tira@node2:~$ ./myprogram.sh -p1 42 -p2 "method2"
```

6

Parameter 1	Parameter 2	Output Directory	Performance
42	Method1	output-directory	0.89
42	Method2	output-directory	0.71

Architecture

Design Goals: Existing Experimentation Frameworks

| Tool	URL	Domain	1	2	3	4	5
evaluatIR	`www.evaluatir.org`	IR	✗	✓	✓	✓	✗
expDB	`expdb.cs.kuleuven.be`	ML	✗	✗	✗	✓	✗
MLComp	`www.mlcomp.org`	ML	✗	✓	✗	✓	✗
myExperiment	`www.myexperiment.org`	any	✗	✓	✓	✓	✗
NEMA	`www.music-ir.org`	IR	✗	✓	✗	✓	✗
TunedIT	`www.tunedit.org`	ML, DM	✓	✓	✗	✓	✗
Yahoo Pipes	`pipes.yahoo.com`	Web	✗	✓	✗	✗	✗

(1) Local instantiation

(2) Web dissemination

(3) Platform independence

(4) Result retrieval

(5) Peer-to-peer collaboration

Architecture

"Experiments as a Service"

Architecture

"Experiments as a Service"
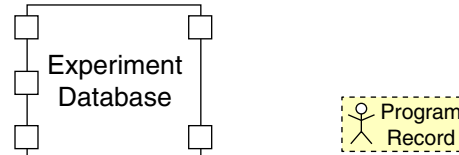
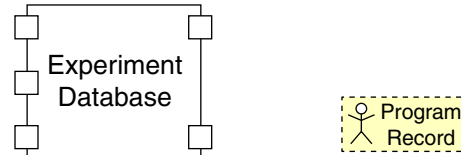| Front-end process	Back-end process

ProgramRecord

❑ A JSON-based program deployment descriptor. Example:

```
{
"MAIN":  "java -jar websearch.jar '$Query' $Results $Engine",
"Results":[1,10,100],
"Query":".+",
"Engine":["CHATNOIR","WIKIPEDIA","BING","GOOGLE"]
}
```

Architecture

"Experiments as a Service"

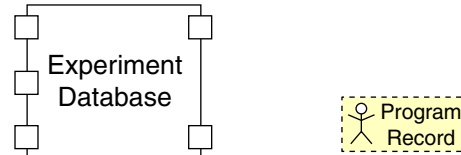| Front-end process	Back-end process

ProgramRecord

❑ A JSON-based program deployment descriptor. Example:

```
{
"MAIN":  "java -jar websearch.jar '$Query' $Results $Engine",
"Results":[1,10,100],
"Query":".+",
"Engine":["CHATNOIR","WIKIPEDIA","BING","GOOGLE"]
}
```


- Results [10] [+]
- Query [tira] [+]
- Engine [CHATNOIR] [+]

Architecture

"Experiments as a Service"

ProgramRecord

❑ A JSON-based program deployment descriptor. Example:

```
{
"MAIN":  "java -jar websearch.jar '$Query' $Results $Engine",
"Results":[1,10,100],
"Query":".+",
"Engine":["CHATNOIR","WIKIPEDIA","BING","GOOGLE"]
}
```


ExperimentDatabase

❑ Stores completed as well as pending experiments.

❑ Indexes the input parameters and provides basic retrieval functionality.

Architecture

"Experiments as a Service"
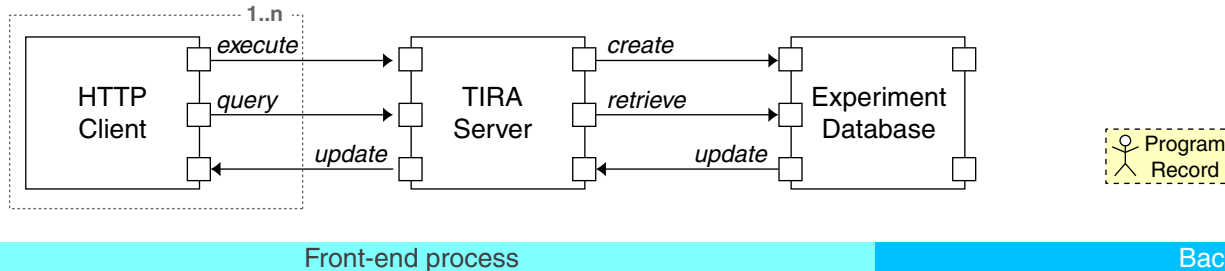
TiraServer

- ❑ Retrieves experiments based on (partial) experiment query.
- ❑ Requests execution of experiment series based on query.
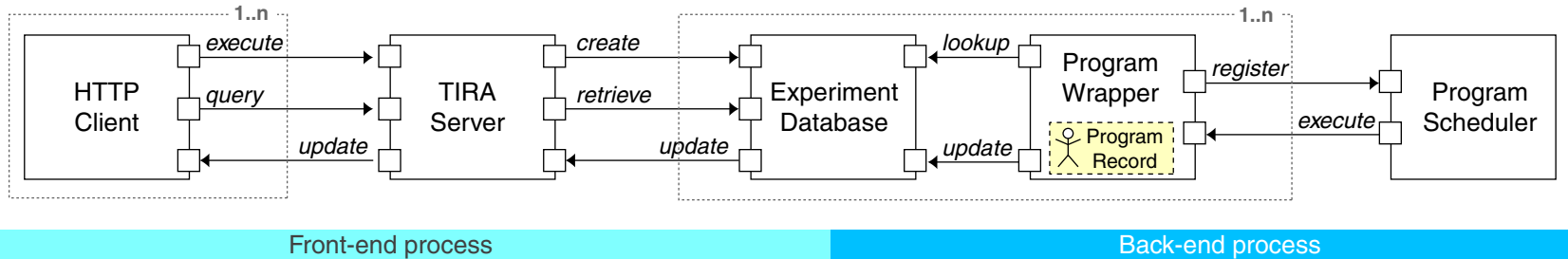- ❑ Realizes web abstraction and creation of TIRA *networks*.

HttpClient

- ❑ Either a Web browser, a client program using the TIRA API, or a remote TiraServer.
- ➜ Can access program-specific information.

Architecture

"Experiments as a Service"

ProgramWrapper

- ❏ Continuously queries the ExperimentDatabase for pending experiments.
- ❏ Registers matching experiments with the ProgramScheduler execution queue.
- ❏ Updates the ExperimentDatabase with notifications and results.

ProgramScheduler

- ❏ Maintains a pool of system threads.
- ❏ Requests execution of the next experiments in the queue.

Architecture

"Experiments as a Service"
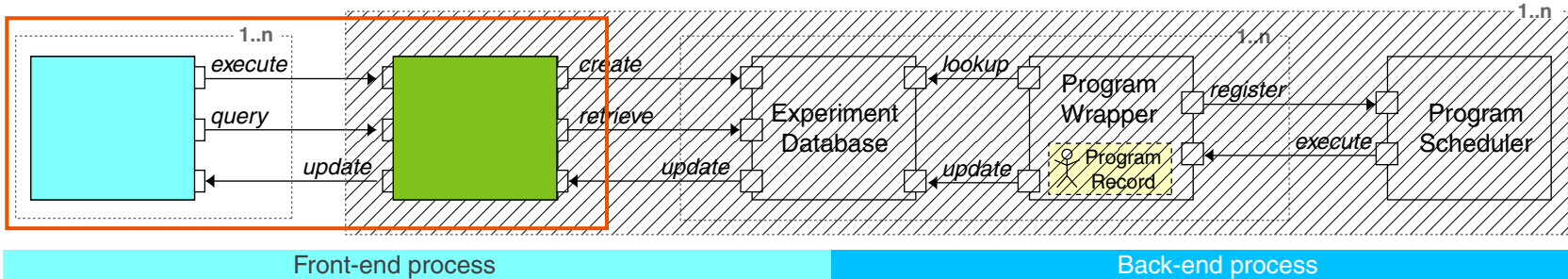
A TIRA network:
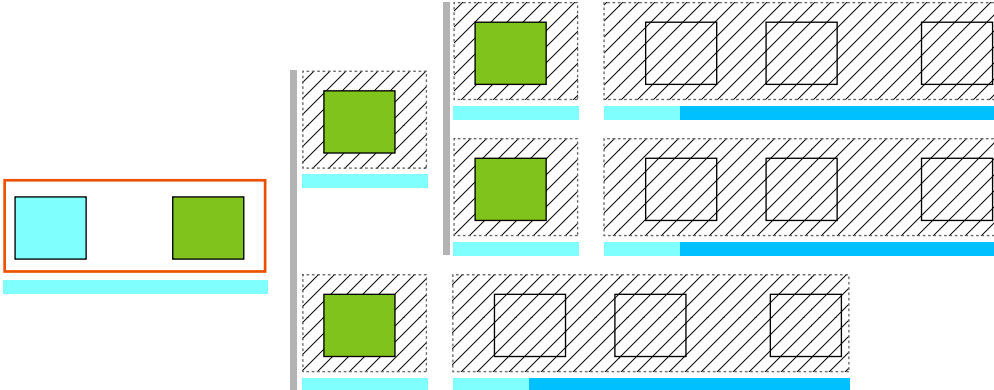
Case Studies

PAN 2012

PAN is a competition on plagiarism detection hosted at CLEF. [pan@clef]

❑ Detailed comparison subtask:

"Given a pair of suspicious and source document,
record all passages in the suspicious document
that are plagiarized from the source document."

❑ Evaluation metric is the *plagdet* score:

$$plagdet(Det, Truth) = \frac{F_1(Det, Truth)}{\log_2(1 + granularity(Det, Truth))}$$

❑ TIRA has been used for the training and evaluation phases.

Case Studies

PAN 2012 – Training Phase

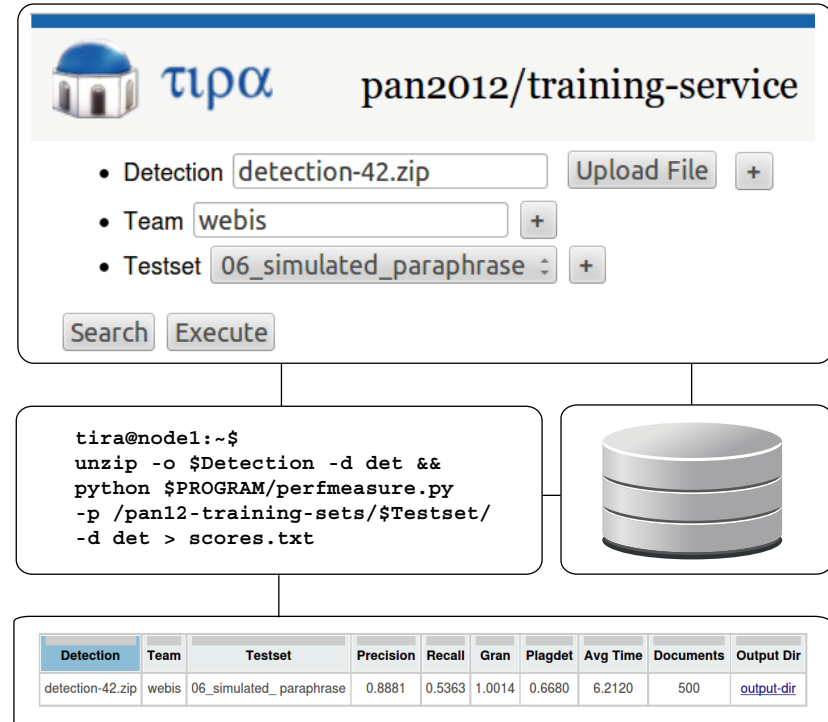- Participants upload detection results for a specific training set.

- From the user inputs the program execution command is generated through substitution.

- Detection results are unzipped and evaluated with an implementation of *plagdet*.

- Participants receive performance results in a result table.

- The training service served as a leaderboard during the competition.

τιρα pan2012/training-service

- Detection detection-42.zip [Upload File] [+]
- Team webis [+]
- Testset 06_simulated_paraphrase ⬍ [+]

[Search] [Execute]

```
tira@node1:~$
unzip -o $Detection -d det &&
python $PROGRAM/perfmeasure.py
-p /pan12-training-sets/$Testset/
-d det > scores.txt
```

Detection	Team	Testset	Precision	Recall	Gran	Plagdet	Avg Time	Documents	Output Dir
detection-42.zip	webis	06_simulated_ paraphrase	0.8881	0.5363	1.0014	0.6680	6.2120	500	output-dir

Case Studies

PAN 2012 – Evaluation Phase

- ❏ TIRA servers are provided for two operating systems, Windows and Ubuntu.

- ❏ Participants submit their plagiarism detection software for deployment on the appropriate TIRA server.

- ❏ A third TIRA server controls the overall evaluation of all deployed submissions on the private test set and provides the overall results.

τιρα pan2012/evaluation-service

- testset 08_all
- detector kongleilei12 +
- detector torrejon12 −
- detector palkovskii12 −

Search   Execute

Windows7   Ubuntu12.04

Testset	Detector	Precision	Recall	Gran	Plagdet	Avg Time	Documents	Output Dir
08_all	kongleilei12	0.8249	0.6782	1.0109	0.7386	5.9187	3033	output-dir
08_all	kasprzak12	0.8931	0.5524	1.0000	0.6826	5.3679	3033	output-dir
08_all	torrejon12	0.8344	0.5004	1.0009	0.6252	0.1900	3033	output-dir

[tira@localhost] [tira@buw]

Case Studies

Search Result Clustering

❑ **Task.** Group the ranked lists from search results into coherent clusters to reduce human effort. [Stein et al., 2012]

❑ **Benefit.** Fetch search results from multiple search engines for storage as static resources and reusable assets.

Simulation Data Mining

❑ **Task.** Pre-compute structural design behavior through learning from large volumes of existing simulation results. [Burrows et al., 2011]

❑ **Benefit.** Easily walk through large parameter spaces and avoid duplication of system simulations.

Summary

Initial versions of TIRA:

- Keep it simple.

- System independence is a key requirement.

TIRA at PAN 2012:

- Create more incentives to use TIRA as a leaderboard.

- The powerful parameter-substitution mechanism made it easy to get valid PAN software submissions running.

For the future:

- Automated program deployment, e.g. Google App Engine.

- Move from open source to open development.

Summary

1. A clear need exists for a community evaluation service.

2. An ideal solution should consider local instantiation, platform independence, result retrieval, web dissemination, and peer-to-peer collaboration.

3. None of the existing solutions meet all of these goals.

4. The TIRA solution is "Experiments as a Service", which takes a locally executable program and transforms it into a web service.

5. TIRA was applied at PAN 2012 with success on the detailed comparison plagiarism detection task.

6. TIRA will be further developed in the future for evaluation initiatives and fostering other collaborations.

Summary

1. A clear need exists for a community evaluation service.

2. An ideal solution should consider local instantiation, platform independence, result retrieval, web dissemination, and peer-to-peer collaboration.

3. None of the existing solutions meet all of these goals.

4. The TIRA solution is "Experiments as a Service", which takes a locally executable program and transforms it into a web service.

5. TIRA was applied at PAN 2012 with success on the detailed comparison plagiarism detection task.

6. TIRA will be further developed in the future for evaluation initiatives and fostering other collaborations.

Thank you!