Package 'dief'

October 13, 2022

Type Package

Title Metrics for Continuous Efficiency

Version 1.2

Date 2019-02-26

Author Maribel Acosta

Maintainer Maribel Acosta <maribel.acosta@kit.edu>

Description An implementation of the metrics dief@t and dief@k to measure the diefficiency (or continuous efficiency) of incremental approaches, see Acosta, M., Vidal, M. E., & Sure-Vetter, Y. (2017) <doi:10.1007/978-3-319-68204-4_1>. The metrics dief@t and dief@k allow for measuring the diefficiency during an elapsed time period t or while k answers are produced, respectively. dief@t and dief@k rely on the computation of the area under the curve of answer traces, and thus capturing the answer rate concentration over a time interval.

License MIT + file LICENSE

Imports flux, fmsb, ggplot2, plyr, graphics, utils

LazyData true

URL https://github.com/maribelacosta/dief

BugReports https://github.com/maribelacosta/dief/issues

RoxygenNote 6.1.1

NeedsCompilation no

Repository CRAN

Date/Publication 2019-02-28 10:10:03 UTC

R topics documented:

`dief` *Tools for Computing Diefficiency Metrics*

Description

An implementation of the metrics dief@t and dief@k to measure the diefficiency (or continuous efficiency) of incremental approaches, see Acosta, M., Vidal, M. E., & Sure-Vetter, Y. (2017) <doi:10.1007/978-3-319-68204-4_1>. The metrics dief@t and dief@k allow for measuring the diefficiency during an elapsed time period t or while k answers are produced, respectively. dief@t and dief@k rely on the computation of the area under the curve of answer traces, and thus capturing the answer rate concentration over a time interval.

Details

Package:	dief
Type:	Package
Version:	1.2
Date:	2017-10-30
License:	MIT

Author(s)

Maribel Acosta

Maintainer: Maribel Acosta <maribel.acosta@kit.edu>

References

Maribel Acosta, Maria-Esther Vidal, and York Sure-Vetter. "Diefficiency metrics: Measuring the continuous efficiency of query processing approaches." In International Semantic Web Conference, pp. 3-19. Springer, Cham, 2017.

Examples

```
# This example uses the answer traces provided in the package.
```

```
# These traces record the answers produced by three approaches "Selective",
# "Not Adaptive", "Random"  when executing the test "Q9.sparql"
data(traces)

# Plot answer traces for test "Q9.sparql"
plotAnswerTrace(traces, "Q9.sparql")

# Compute dief@t with t the time where the slowest approach produced the last answer.
dieft(traces, "Q9.sparql")

# Compute dief@t after 7.5 time units (seconds) of execution.
dieft(traces, "Q9.sparql", 7.5)
```

`diefk` *Compute metric dief@k*

Description

This function computes the dief@k metric at a given k (number of answers).

Usage

```
diefk(inputtrace, inputtest, k = -1)
```

Arguments

`inputtrace`	dataframe with the answer trace. Attributes of the dataframe: test, approach, answer, time.
`inputtest`	string that specifies the specific test to analyze from the answer trace.
`k`	number of answers to compute diefk. By default, the function computes the minimum of the total number of answers produced by the approaches.

Author(s)

Maribel Acosta

See Also

dieft, diefk2, plotAnswerTrace

Examples

```
# Compute dief@k when k is the number of answers produced
# by the approach theat generated the least answers.
diefk(traces, "Q9.sparql")
# Compute dief@k while producing the first k=1000 answers.
diefk(traces, "Q9.sparql", 1000)
```

diefk2 *Compute dief@k at a portion of the answer*

Description

This function computes the dief@k metric at a given kp (portion of answers).

Usage

```
diefk2(inputtrace, inputtest, kp = -1)
```

Arguments

inputtrace	dataframe with the answer trace. Attributes of the dataframe: test, approach, answer, time.
inputtest	string that specifies the specific test to analyze from the answer trace.
kp	portion of answers to compute diefk (between 0.0 and 1.0). By default and when kp=1.0, this function behaves the same as diefk. It computes the kp portion of of minimum of of number of answers produced by the approaches.

Author(s)

Maribel Acosta

See Also

dieft, diefk, plotAnswerTrace

Examples

```
# Compute dief@k when the approaches produced 25% of the answers w.r.t.
# the approach that produced the least answers.
diefk2(traces, "Q9.sparql", 0.25)
```

dieft *Compute metric dief@t*

Description

This function computes the dief@t metric at a point in time t.

Usage

```
dieft(inputtrace, inputtest, t = -1)
```

Arguments

inputtrace	dataframe with the answer trace. Attributes of the dataframe: test, approach, answer, time.
inputtest	string that specifies the specific test to analyze from the answer trace.
t	point in time to compute dieft. By default, the function computes the minimum of the execution time among the approaches in the answer trace.

Author(s)

Maribel Acosta

See Also

diefk, diefk2, plotAnswerTrace

Examples

```
# Compute dief@t when t is the time where the slowest approach produced the last answer.
dieft(traces, "Q9.sparql")
# Compute dief@t after 7.5 time units (seconds) of execution.
dieft(traces, "Q9.sparql", 7.5)
```

---

experiment1 *Compares dief@t with other benchmark metrics as in <doi:10.1007/978-3-319-68204-4_1>*

Description

This function repeats the results reported in "Experiment 1" in Acosta, M., Vidal, M. E., & Sure-Vetter, Y. (2017) <doi:10.1007/978-3-319-68204-4_1>. Experiment 1 compares the performance of querying approaches when using metrics defined in the literature (total execution time, time for the first tuple, throughput, and completeness) and the metric dieft@t.

Usage

```
experiment1(traces, metrics)
```

Arguments

traces	dataframe with the result of the traces. The structure of this dataframe is as follows: "test,approach,tuple,time".
metrics	dataframe with the result of the other metrics. The structure of this dataframe is as follows: "test,approach,tfft,totaltime,comp".

Author(s)

Maribel Acosta

See Also

experiment2, dieft

Examples

```
# To fully reproduce the experiments download the full files and load them using read.csv:
# traces is available at <https://figshare.com/files/9625852>
# metrics is available at <https://figshare.com/files/9660316>
results1 <- experiment1(traces, metrics)
```

`experiment2` *Compares dief@k at different answer portions as in <doi:10.1007/978-3-319-68204-4_1>*

Description

This function repeats the results reported in Experiment 2 in Acosta, M., Vidal, M. E., & Sure-Vetter, Y. (2017) <doi:10.1007/978-3-319-68204-4_1>. "Experiment 2" measures the continuous efficiency of approaches when producing the first 25

Usage

```
experiment2(traces)
```

Arguments

`traces`	dataframe with the result of the traces. The structure of this dataframe is as follows: "test,approach,tuple,time".

Author(s)

Maribel Acosta

See Also

experiment1, diefk2

Examples

```
# To fully reproduce the experiments download the full file and load it using read.csv:
# traces is available at <https://figshare.com/files/9625852>
results2 <- experiment2(traces)
```

metrics *Example of benchmarking performance with other metrics*

Description

A dataset with the results of measuring the performance of three approaches with four metrics. The variables are as follows:

Usage

```
data(metrics)
```

Format

A data frame with 3 rows and 5 variables

Details

- test: id of the test (in this case a SPARQL query) executed. Example: 'Q9.sparql'.
- approach: name of the approach (or engine) used to execute the query.
- tfft: time (in seconds) required by approach to produce the first tuple when executing query.
- totaltime: elapsed time (in seconds) since approach started the execution of query until the answer i is produced.
- comp: number of answers produced by approach when executing query.

Source

nLDE SPARQL engine: computing diefficiency metrics based on answer traces and query processing performance benchmarking

plotAnswerTrace *Plot the answer trace of approaches*

Description

This function plots the answer trace of the approaches when executing a given test.

Usage

```
plotAnswerTrace(inputtrace, inputtest)
```

Arguments

inputtrace	dataframe with the answer trace. Attributes of the dataframe: test, approach, answer, time.
inputtest	string that specifies the specific test to analyze from the answer trace.

Author(s)

Maribel Acosta

See Also

diefk, dieft

Examples

```
plotAnswerTrace(traces, "Q9.sparql")
```

`plotExperiment1` *Generate radar plots that compare dief@t with other benchmark metrics in all tests as in <doi:10.1007/978-3-319-68204-4_1>*

Description

This function plots the results reported in Experiment 1 in Acosta, M., Vidal, M. E., & Sure-Vetter, Y. (2017) <doi:10.1007/978-3-319-68204-4_1>. Experiment 1 compares the performance of querying approaches when using metrics defined in the literature (total execution time, time for the first tuple, throughput, and completeness) and the metric dieft@t.

Usage

```
plotExperiment1(allmetrics)
```

Arguments

`allmetrics`	dataframe with the result of all the metrics in Experiment 1.

Author(s)

Maribel Acosta

See Also

experiment1, diefk2 results1 <- experiment1(traces, metrics) plotExperiment1(results1)

`plotExperiment1Test` *Generate radar plots that compare dief@t with other benchmark metrics in a specific test as in <doi:10.1007/978-3-319-68204-4_1>*

Description

This function plots the results reported for a single given test in "Experiment 1" in Acosta, M., Vidal, M. E., & Sure-Vetter, Y. (2017) <doi:10.1007/978-3-319-68204-4_1>. Experiment 1 compares the performance of querying approaches when using metrics defined in the literature (total execution time, time for the first tuple, throughput, and completeness) and the metric dieft@t.

Usage

```
plotExperiment1Test(allmetrics, q)
```

Arguments

`allmetrics`	dataframe with the results of all the metrics in Experiment 1.
`q`	id of the selected test to plot.

Author(s)

Maribel Acosta

See Also

experiment1, plotExperiment1

Examples

```
results1 <- experiment1(traces, metrics)
plotExperiment1Test(results1, "Q9.sparql")
```

`plotExperiment2` *Generate radar plots that compare dief@k at different answer completeness in all tests as in <doi:10.1007/978-3-319-68204-4_1>*

Description

This function plots the results reported in Experiment 2 in Acosta, M., Vidal, M. E., & Sure-Vetter, Y. (2017) <doi:10.1007/978-3-319-68204-4_1>. "Experiment 2" measures the continuous efficiency of approaches when producing the first 25

Usage

```
plotExperiment2(diefkDF)
```

Arguments

diefkDF	dataframe with the results of Experiment 2.

Author(s)

Maribel Acosta

See Also

experiment2, diefk2

Examples

```
results2 <- experiment2(traces)
plotExperiment2(results2)
```

---

plotExperiment2Test *Generate radar plots that compare dief@k at different answer completeness in a specific test as in <doi:10.1007/978-3-319-68204-4_1>*

Description

This function plots the results reported for a single given test in "Experiment 2" in Acosta, M., Vidal, M. E., & Sure-Vetter, Y. (2017) <doi:10.1007/978-3-319-68204-4_1>. "Experiment 2" measures the continuous efficiency of approaches when producing the first 25

Usage

```
plotExperiment2Test(diefkDF, q)
```

Arguments

diefkDF	dataframe resulting from Experiment 2.
q	id of the selected test to plot.

Author(s)

Maribel Acosta

See Also

experiment2, plotExperiment2

Examples

```
results2 <- experiment2(traces)
plotExperiment2Test(results2, "Q9.sparql")
```

`traces` *Example of answer traces*

Description

A dataset containing answer traces of executing three approaches. The variables are as follows:

Usage

```
data(traces)
```

Format

A data frame with 1543 rows and 4 variables

Details

- test: id of the test (in this case a SPARQL query) executed. Example: 'Q9.sparql'.
- approach: name of the approach (or engine) used to execute the query.
- answer: the value i indicates that this row corresponds to the ith answer produced by approach when executing query.
- time: elapsed time (in seconds) since approach started the execution of query until the answer i is produced.

Source

nLDE SPARQL engine: computing diefficiency metrics based on answer traces and query processing performance benchmarking

Index