
ir-kit Documentation

Release

Harry Scells

Jun 30, 2017

Contents:

1	Trec	1
1.1	Qrels	1
1.2	Runs	2
1.3	Results	3
2	Query	5
2.1	Elasticsearch	5
3	Plot	7
4	Installing	9
5	Usage	11
5.1	Command line tools:	11
5.2	Libraries	11
6	Indices and tables	13
	Python Module Index	15

Qrels

Functions and classes for dealing with English qrel files. The specification can be viewed at: http://trec.nist.gov/data/qrels_eng/

Usage:

```

>>> sample_qrels = '1 0 AP880212-0161 0\n1 0 AP880216-0139 1\n1 0 AP880216-0169 0'
>>> qrels = loads(sample_qrels)
>>> qrels.topic
['1', '1', '1']
>>> [x.topic for x in qrels['1']]
['1', '1', '1']

```

Harry Scells Mar 2017

class `irkit.trec.qrels.Qrel` (*topic: str, iteration: int, document_num: str, relevancy: int*)

A line in a qrels file conforming to the specification at: http://trec.nist.gov/data/qrels_eng/

class `irkit.trec.qrels.Qrels` (*qrels: typing.List[irkit.trec.qrels.Qrel]*)

A python representation of a qrels file. It is a list of Qrel objects.

dump (*fp: _io.TextIOWrapper*) → None

Dump the qrels to a file

Parameters *fp* – A File pointer

dumps () → str

Dump the qrels to a string.

Returns Formatted qrels

`irkit.trec.qrels.load` (*qrels: _io.TextIOWrapper*) → `irkit.trec.qrels.Qrels`

Load qrels from a file.

Parameters *qrels* – File pointer

Returns Qrels object

`irkit.trec.qrels.loads(qrels: str) → irkit.trec.qrels.Qrels`
 Load qrels from a string.

Parameters `qrels` – Some string representation of qrels

Returns Qrels object

Runs

Functions and classes for dealing with trec_eval run files. For a description on how run files look, see: http://faculty.washington.edu/levow/courses/ling573_SPR2011/hw/trec_eval_desc.htm

Usage:

```
>>> sample_run = '''351 0 DOC1 1 100 run-name\n351 0 DOC2 2 50 run-
↳ name'''
>>> runs = loads(sample_run)
>>> runs.dumps()
'351 Q0 DOC1 1 100 run-name
351 Q0 DOC2 2 50 run-name'
>>> [str(run) for run in runs.runs]
['351 Q0 DOC1 1 100 run-name', '351 Q0 DOC2 2 50
↳ run-name']
>>> [str(run) for run in runs['351']]
['351 Q0 DOC1 1 100 run-name', '351 Q0 DOC2 2 50
↳ run-name']
>>> runs.rank
['1', '2']
>>> TrecEvalRuns(runs['351']).rank
['1', '2']
```

Harry Scells May 2017

class `irkit.trec.run.TrecEvalRun(topic: str, q: int, doc_id: str, rank: int, score: float, run_id: str)`
 TrecEvalRun is a container class for a line in a trec_eval run file.

class `irkit.trec.run.TrecEvalRuns(runs: typing.List[irkit.trec.run.TrecEvalRun])`
 TrecEvalRuns is a wrapper around a TrecEvalRun which is just a container class for a line in a trec_eval run file. This class contains some convenience functions for dealing with runs, such as getting a list of the runs but topic id or slicing by column.

dump (`fp: _io.TextIOWrapper`) → None
 Dump the qrels to a file

Parameters `fp` – A File pointer

dumps () → str
 Dump the qrels to a string.

Returns Formatted qrels

`irkit.trec.run.load(runs: _io.TextIOWrapper) → irkit.trec.run.TrecEvalRuns`
 Load a trec_eval run file.

Parameters `runs` – A file pointer containing runs.

Returns TrecEvalRuns

`irkit.trec.run.loads (runs: str) → irkit.trec.run.TrecEvalRuns`

Load a trec_eval run file from a string.

Parameters `runs` – A string containing runs.

Returns TrecEvalRuns

Results

Functions and classes for dealing with trec_eval results files.

Harry Scells Mar 2017

class `irkit.trec.results.TrecEvalResults (run_id: str, results: typing.Dict, queries: typing.Dict)`

An object that stores the results output by trec_eval.

`irkit.trec.results.load (trec_result_file: _io.TextIOWrapper) → irkit.trec.results.TrecEvalResults`

Load trec_eval results from a file.

Parameters `trec_result_file` – File pointer

Returns Qrels object

`irkit.trec.results.loads (trec_results: str) → irkit.trec.results.TrecEvalResults`

Load trec_eval results from a string.

Parameters `trec_results` – Some string representation of trec results

Returns TrecEvalResults object

Elasticsearch

Functions and classes for dealing with Elasticsearch queries.

Harry Scells Jun 2017

class `irkit.query.elasticsearch.Visitor` (*node_name: str*)

Visitor interface for performing analysis on an Elasticsearch query. Implement the visit method for the node to perform some action. A visitor can either store a result to be returned by the traverse function, or

visit (*node: dict*)

Implement this class to visit nodes in an Elasticsearch query. When implementing for traverse, place the result in `self.result`. When implementing for transform, return the transformed node as a dict.

Parameters `node` – A node in the Elasticsearch query.

Returns See description.

`irkit.query.elasticsearch.transform` (*query: dict, visitor: irkit.query.elasticsearch.Visitor*) → dict

Transform the query using a visitor. Visitors used by this function modify the query in-place and should not return anything. The following example will transform all of the must queries to must_not queries:

```
>>> class ExampleVisitor(Visitor):
...     def __init__(self, node_name: str):
...         super().__init__(node_name)
...
...     def visit(self, node: dict):
...         node['must_not'] = node[self.node_name]
...         del node[self.node_name]
>>> visitor = ExampleVisitor('must')
>>> transform({'query': {'must': {'match': 'example'}}}, visitor)
{'query': {'must_not': {'match': 'example'}}
```

Parameters

- **query** – An ElasticSearch query.
- **visitor** – An implemented Visitor class.

Returns A modified query

`irkit.query.elasticsearch.traverse(query: dict, visitor: irkit.query.elasticsearch.Visitor) → typing.Generic`

Traverse down the query tree using the specified visitor. Visitors used by this function cannot modify the query. Instead they must store their return value into the result value on the visitor. This is useful for getting statistics about a query. The following example extracts all of the keywords in match queries:

```
>>> class ExampleVisitor(Visitor):
...     def __init__(self, node_name: str):
...         super().__init__(node_name)
...         self.result = []
...
...     def visit(self, node: dict):
...         self.result.append(node[self.node_name])
>>> visitor = ExampleVisitor('match')
>>> traverse({'query': {'must': {'match': 'example'}}}, visitor)
['example']
```

Parameters

- **query** – An ElasticSearch query.
- **visitor** – An implemented Visitor class.

Returns The value stored in visitor.result.

Various plotting functions.

Harry Scells Mar 2017

```
irkit.plot.trecplot.pr_curve(results: typing.List[irkit.trec.results.TrecEvalResults])
                                → <module 'matplotlib.pyplot' from
                                '/home/docs/checkouts/readthedocs.org/user_builds/ir-
                                kit/envs/latest/local/lib/python3.5/site-packages/matplotlib-2.0.2-
                                py3.5-linux-x86_64.egg/matplotlib/pyplot.py'>
```

Create a precision-recall graph from trec_eval results.

Parameters **results** – A list of TrecEvalResults files.

Returns a matplotlib plt object

```
irkit.plot.trecplot.topic_ap(results: typing.List[irkit.trec.results.TrecEvalResults],
                             sort_on_ap=False)
```

Create an average-precision topic visualisation.

Parameters

- **results** – A list of TrecEvalResults files.
- **sort_on_ap** – Should the visualisation be sorted using average precision?

Returns a matplotlib plt object.

CHAPTER 4

Installing

IR Kit can be installed via pip:

```
pip3 install ir-kit
```


Command line tools:

For generating precision-recall curves and plotting the average precision of a topic there is `trecplot`:

```
trecplot --help
```

Libraries

Dealing with trec-related files is done using the `trec` package. This package contains classes for dealing with qrel files, trec run files, and trec result files.

CHAPTER 6

Indices and tables

- `genindex`
- `modindex`
- `search`

i

- `irkit.plot`, 7
- `irkit.plot.trecplot`, 7
- `irkit.query`, 5
- `irkit.query.elasticsearch`, 5
- `irkit.trec`, 1
- `irkit.trec.qrels`, 1
- `irkit.trec.results`, 3
- `irkit.trec.run`, 2

D

`dump()` (irkit.trec.qrels.Qrels method), 1
`dump()` (irkit.trec.run.TrecEvalRuns method), 2
`dumps()` (irkit.trec.qrels.Qrels method), 1
`dumps()` (irkit.trec.run.TrecEvalRuns method), 2

I

`irkit.plot` (module), 7
`irkit.plot.trecplot` (module), 7
`irkit.query` (module), 5
`irkit.query.elasticsearch` (module), 5
`irkit.trec` (module), 1
`irkit.trec.qrels` (module), 1
`irkit.trec.results` (module), 3
`irkit.trec.run` (module), 2

L

`load()` (in module irkit.trec.qrels), 1
`load()` (in module irkit.trec.results), 3
`load()` (in module irkit.trec.run), 2
`loads()` (in module irkit.trec.qrels), 2
`loads()` (in module irkit.trec.results), 3
`loads()` (in module irkit.trec.run), 2

P

`pr_curve()` (in module irkit.plot.trecplot), 7

Q

`Qrel` (class in irkit.trec.qrels), 1
`Qrels` (class in irkit.trec.qrels), 1

T

`topic_ap()` (in module irkit.plot.trecplot), 7
`transform()` (in module irkit.query.elasticsearch), 5
`traverse()` (in module irkit.query.elasticsearch), 6
`TrecEvalResults` (class in irkit.trec.results), 3
`TrecEvalRun` (class in irkit.trec.run), 2
`TrecEvalRuns` (class in irkit.trec.run), 2

V

`visit()` (irkit.query.elasticsearch.Visitor method), 5
`Visitor` (class in irkit.query.elasticsearch), 5