Streamlining Evaluation with `ir-measures`

Sean MacAvaney, Craig Macdonald, Iadh Ounis

University of Glasgow Terrier

Documentation: **https://ir-measur.es/**
Github: **terrierteam/ir_measures**

A Natural Python library for IR Evaluation

(mean) average precision → `AP(rel=2)@10` ← rank cutoff; min. relevance label

Automatically Delegates Computation to Established Tools

(decoupling semantics and implementation)

`AP(rel=2)@10` —invokes→ via `pytrec_eval`[1]

```
trec_eval qrels run -m map_cut.20 -l2
```

rank biased precision → `RBP(p=0.3,rel=2)` —invokes→ via `cwl_eval`[2]

```
cwl_eval.ruler.RBPCWLMetric(0.3)
```

(+ other handling for rel=2)

persistence

Also supports measures from gdeval, msmarco-eval, ndeval, ranx, trectools, and others:

`alpha_nDCG, Accuracy, AP_IA, BPM, Bpref, Compat, ERR@k, ERR_IA, infAP, INSQ, INST, IPrec, Judged@k, MAP, nDCG, NERR10, NERR11, NERR8, NERR9, nERR_IA, nNRBP, NRBP, NumQ, NumRel, NumRet, P@k, P_IA, R@k, RBP, Rprec, RR, SDCG@k, SetAP, SetF, SetP, SetR, StRecall@k, Success@k`

Several Convenient Interfaces

Example: `ir_measures` computing TREC DL's official measures:

(With `trec_eval`, requires multiple invocations)

```
$ trec_eval qrels run -m ndcg_cut.10
ndcg_cut_10 0.5536

$ trec_eval qrels run -m map -m recip_rank -l2
recip_rank  0.6996
map 0.3684
```

Old way: trec_eval with multiple invocations

Command Line

```
$ ir_measures qrels run 'nDCG@10 RR(rel=2) AP(rel=2)'
nDCG@10     0.5536
RR(rel=2)   0.6996
AP(rel=2)   0.3684
```

single invocation for measures with different settings

Python

```
ir_measures.calc_aggreate([nDCG@10 RR(rel=2) AP(rel=2)], qrels, run)
{
  nDCG@10: 0.5536,
  RR(rel=2): 0.6996,
  AP(rel=2): 0.3684
}
```

natural expression of measures, right in Python

PyTerrier [3]

using ir-measures

```
pt.Experiment(
  [pipeline1, pipeline2],
  dataset.get_topics(), dataset.get_qrels(),
  [nDCG@10 RR(rel=2) AP(rel=2)],
  baseline=0 # perform stat tests
)
```

perform stat. tests

name	nDCG@10	AP(rel=2)	RR(rel=2)	nDCG@10 p-value	...
pipeline1	0.553616	0.368382	0.699594	NaN	...
pipeline2	0.688357	0.419591	0.840698	0.000581	...

Also Integrated with other tools, including ir-datasets, OpenNIR, Experimaestro, DiffIR, and Patapsco.

Extras

📐 **Explore Measures:**
Make changes to a ranking list and see the effect it has on a set of measures in real-time.
https://demo.ir-measur.es/explore

🔧 **Reverse Measures:**
Find example ranking lists that produce scores for a given measure.
https://demo.ir-measur.es/reverse

References

[1] Van Gysel & de Rijke. Pytrec_eval: An Extremely Fast Python Interface to trec_eval. SIGIR 2018.
[2] Azzopardi et al. Cwl_eval: An Evaluation Tool for Information Retrieval. SIGIR 2019.
[3] Macdonald et al. PyTerrier: Declarative Experimentation in Python from BM25 to Dense Retrieval. CIKM 2021.