The OldDog Docker Image for OSIRRC at SIGIR 2019

Chris Kamphuis and Arjen P. de Vries

iCIS | Data Science
Radboud University

Overview

Document representation in a column store relational database

Premise:

- Express ranking function as SQL queries
- Easy comparison of different ranking functions

The column store relational database

Represent the data in a column store database

I put on my shoes after I put on my socks.

termid	term	df
1	put	2
2	shoes	1
3	after	1
4	socks	1

Table: *dict*

termid	docid	count
1	1	2
2	1	1
3	1	1
4	1	1

Table: *terms*

collection_id	id	len
doc1	1	5

Table: *docs*

Conjunctive BM25

```
WITH qterms AS (SELECT termid, docid, count FROM terms
  WHERE termid IN (591020, 720333, 462570)),
```

Conjunctive BM25

```
WITH qterms AS (SELECT termid, docid, count FROM terms
  WHERE termid IN (591020, 720333, 462570)),
subscores AS (SELECT docs.collection_id , docs.id , len ,
  term_tf.termid, term_tf.tf , df ,
  (log((528030−df+0.5)/(df+0.5))∗((term_tf.tf∗(1.2+1)/
  (term_tf.tf+1.2∗(1−0.75+0.75∗(len/188.33)))))) AS subscore
```

Conjunctive BM25

```
WITH qterms AS (SELECT termid, docid, count FROM terms
  WHERE termid IN (591020, 720333, 462570)),
subscores AS (SELECT docs.collection_id, docs.id, len,
  term_tf.termid, term_tf.tf, df,
  (log((528030−df+0.5)/(df+0.5))*((term_tf.tf*(1.2+1)/
  (term_tf.tf+1.2*(1−0.75+0.75*(len/188.33)))))) AS subscore
FROM (SELECT termid, docid, count as tf FROM qterms) AS term_tf
  JOIN (SELECT docid FROM qterms
    GROUP BY docid HAVING COUNT(distinct termid) = 3)
    AS cdocs ON term_tf.docid = cdocs.docid
  JOIN docs ON term_tf.docid = docs.id
  JOIN dict ON term_tf.termid = dict.termid)
```

Conjunctive BM25

```
WITH qterms AS (SELECT termid, docid, count FROM terms
  WHERE termid IN (591020, 720333, 462570)),
subscores AS (SELECT docs.collection_id, docs.id, len,
  term_tf.termid, term_tf.tf, df,
  (log((528030-df+0.5)/(df+0.5))*((term_tf.tf*(1.2+1)/
  (term_tf.tf+1.2*(1-0.75+0.75*(len/188.33)))))) AS subscore
FROM (SELECT termid, docid, count as tf FROM qterms) AS term_tf
  JOIN (SELECT docid FROM qterms
    GROUP BY docid HAVING COUNT(distinct termid) = 3)
    AS cdocs ON term_tf.docid = cdocs.docid
  JOIN docs ON term_tf.docid = docs.id
  JOIN dict ON term_tf.termid = dict.termid)
SELECT scores.collection_id, score
  FROM (SELECT collection_id, SUM(subscore) AS score
    FROM subscores
    GROUP BY collection_id) AS scores
    JOIN docs ON scores.collection_id =docs.collection_id
  ORDER BY score DESC;
```

Results

Effectiveness scores

	Robust04		Core18	
	MAP	P@30	MAP	P@30
Conjunctive BM25	0.1736	0.2526	0.1802	0.3167
Disjunctive BM25	0.2434	0.2985	0.2381	0.3313

Conclusion

- Effectiveness scores difference between conjunctive and disjunctive is more than expected
  - Robust MAP: 0.070 | -28.64%
  - Robust P@30: 0.046 | -15.04%
  - Core18 MAP: 0.058 | -24.32%
  - Core18 P@30: 0.015 | -4.41%
- The jig framework helps ‘version control’
  - Topic missing