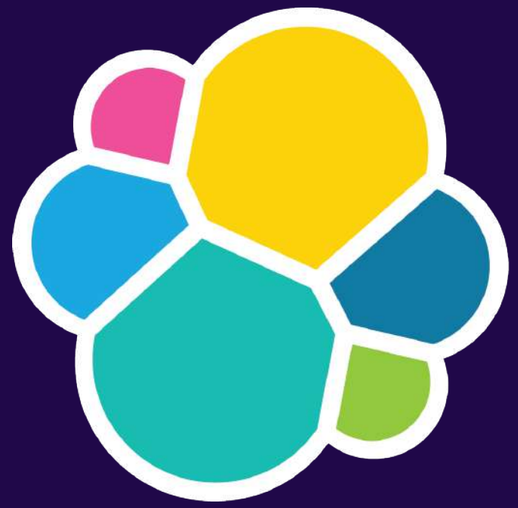
Reciprocal Rank Fusion

How to Stop Worrying about Boosting

Philipp Krenn @xeraa


ARRR
ARRR
F

Who knows what RRF is / does?



elastic

Developer 🥑

@xeraa

Problem: hybrid search — combine multiple search techniques

Solution: normalize scores?!

People frequently want to compute a "Percentage" from Lucene scores to determine what is a "100% perfect" match vs a "50%" match. This is also somethings called a "normalized score"

Don't do this.

Seriously. Stop trying to think about your problem this way, it's not going to end well.

-- https://cwiki.apache.org/confluence/display/LUCENEJAVA/ScoresAsPercentages

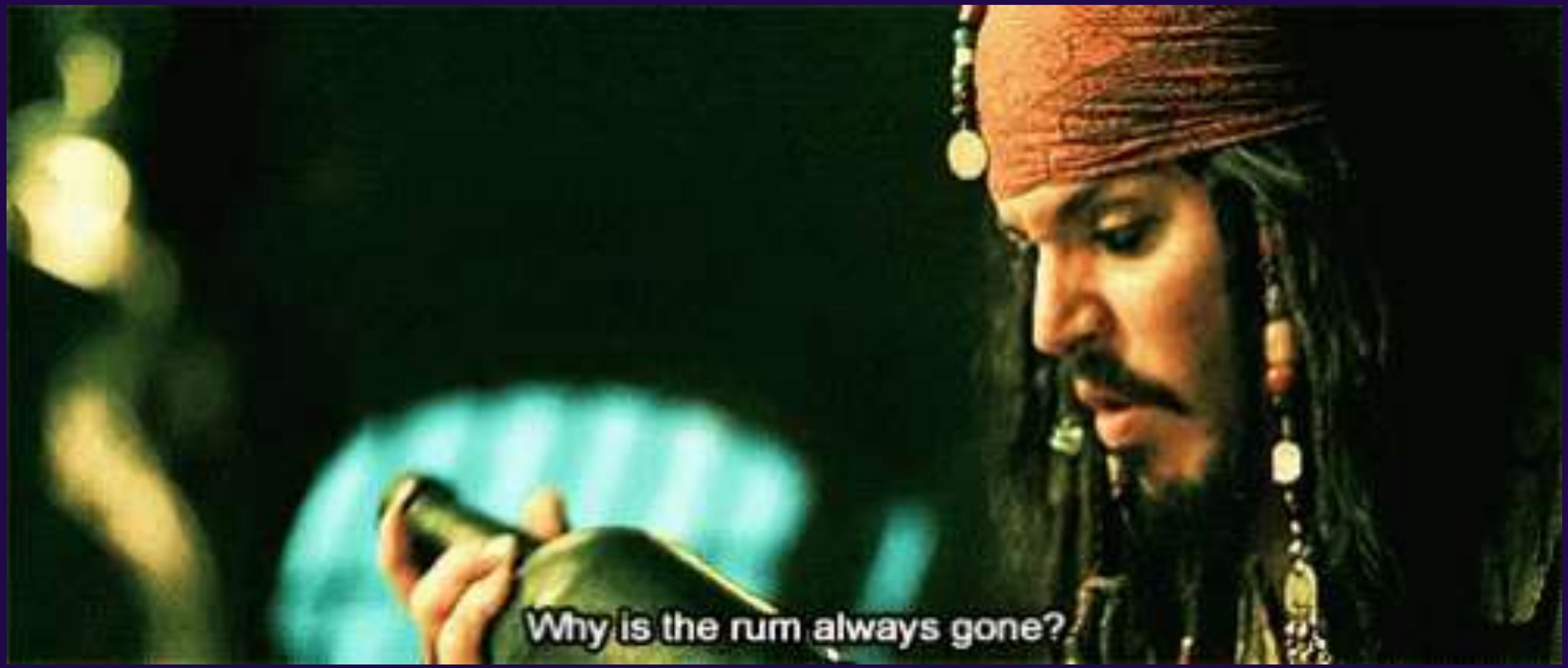
Why?

elastic

@xeraa

elastic

@xeraa


Why is the rum always gone?

```
PUT quotes/_doc/1
{
  "quote": "I prefer rum."
}
PUT quotes/_doc/2
{
  "quote": "But why is the rum gone?!"
}
PUT quotes/_doc/3
{
  "quote": "Why is the rum always gone?"
}
```

```
GET quotes/_search
{
  "query": {
    "match": {
      "quote": "why is the rum always gone"
    }
  }
}
```

```
"max_score": 2.76791
```

== 100%?

```
DELETE quotes/_doc/3

GET quotes/_search
{
  "query": {
    "match": {
      "quote": "why is the rum always gone"
    }
  }
}
```

"max_score": 1.8612611 == 67%?

```
PUT quotes/_doc/4
{
  "quote": "The rum is gone! Why is the rum always gone?"
}

GET quotes/_search
{
  "query": {
    "match": {
      "quote": "why is the rum always gone"
    }
  }
}
```

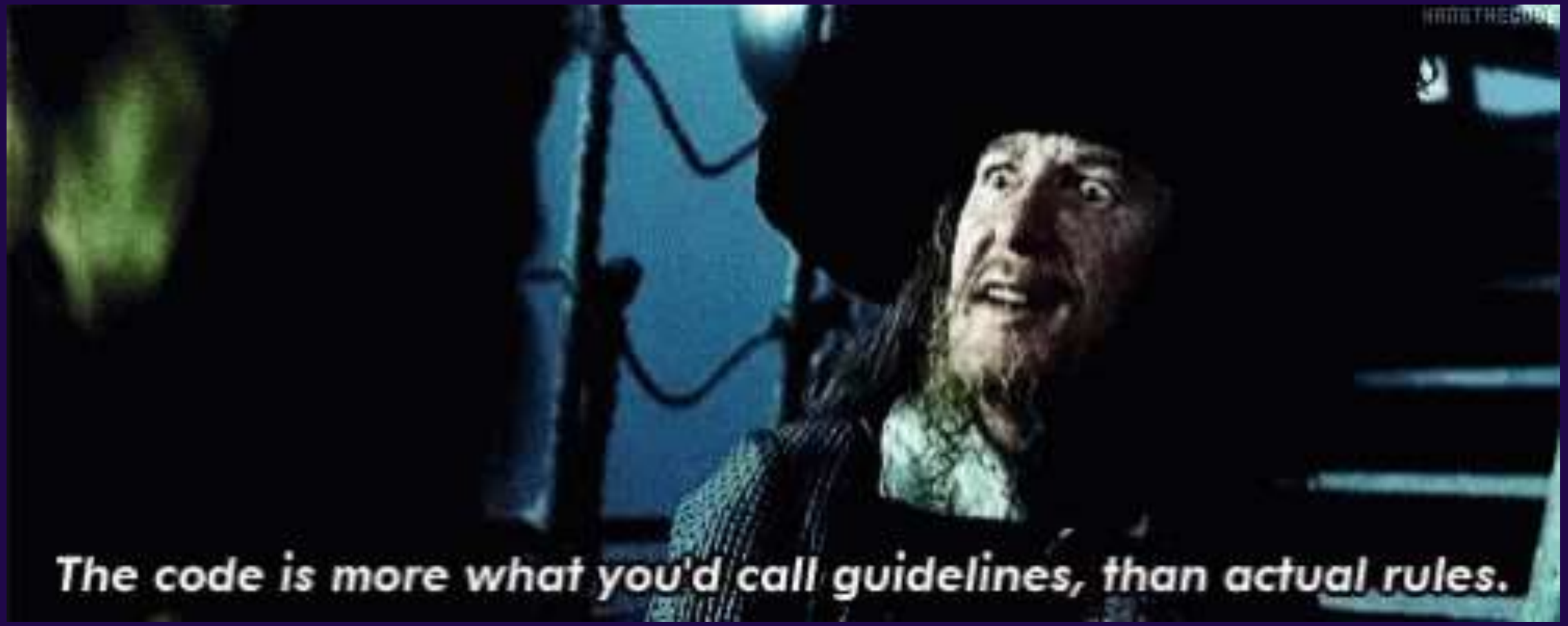
"max_score": 2.9981923 == 108%?


But you have heard of me.
HAHSTHECODE

Solution: boosting?!

Mostly experience

Based on dataset and queries


The code is more what you'd call guidelines, than actual rules.

Solution: max-min scale

Scale each ranking's score to 0-1

RRF Paper

https://plg.uwaterloo.ca/~gvcormac/cormacksigir09-rrf.pdf

Reciprocal Rank Fusion outperforms Condorcet and individual Rank Learning Methods

G. V. Cormack
University of Waterloo
Waterloo, Ontario, Canada

C. L. A. Clarke
University of Waterloo
Waterloo, Ontario, Canada

Stefan Büttcher
Google
Redmond, WA, USA

ABSTRACT

Reciprocal Rank Fusion (RRF), a simple method for combining the document rankings from multiple IR systems, consistently yields better results than any individual system, and better results than the standard method Condorcet Fuse. This result is demonstrated by using RRF to combine the results of several TREC experiments, and to build a meta-learner that ranks the LETOR 3 dataset better than any previously reported method.

lower-ranked documents does not vanish as it would were, say, an exponential function used. The constant k mitigates the impact of high rankings by outlier systems.

Condorcet Fuse combines rankings by sorting the documents according to the pairwise relation $r(d_1) < r(d_2)$, which is determined for each (d_1, d_2) by majority vote among the input rankings. CombMNZ requires for each r a corresponding scoring function $s_r : D \rightarrow \mathbb{R}$ and a cutoff rank c which all contribute to the CombMNZ score:

Step 1: Generate each ranking

Step 2: Calculate the Reciprocal Rank

1st ranked item $\frac{1}{1}$, 2nd $\frac{1}{2}$,...

Step 3: Optional weight or score normalization of rankings

Not part of the original paper

Variants based on recency, score, or trust

Is it a good idea? Kind of back to boosting

Step 4: Sum weighted reciprocal ranks for each item

$$RRF(i) = \sum_{j=1}^{n} w_j * \frac{1}{k + r_{ij}}$$

- $RRF(i)$: Reciprocal Rank Fusion score for item i
- n: number of rankings
- w_j: weight assigned to the j-th ranking (not in paper)
- k: ranking constant (default 60)
- r_{ij}: rank of item i in the j-th ranking

Select *k*

How much influence documents in individual rankings have over the final result set

The higher the value, the more influence lower ranked documents have

Select window size

The size of the individual result sets per query

Higher values will improve result relevance at the cost of performance — for example 100 for top 10 documents

Step 5: Sort all items by Reciprocal Rank Fusion score

Example

Ranking	BM25	BM25 (boosted)	ELSER
1	doc 2	doc 3	doc 4
2	doc 3	doc 5	doc 2
3	doc 5	doc 2	doc 5
4	doc 1	doc 1	doc 3
5	doc 4	doc 4	doc 1

Example

Document	Reciprocal Rank (k=1)	Sum
doc 1	1/(1+4) + 1/(1+4) + 1/(1+5)	0.57
doc 2	1/(1+1) + 1/(1+3) + 1/(1+2)	1.08
doc 3	1/(1+2) + 1/(1+1) + 1/(1+4)	1.03
doc 4	1/(1+5) + 1/(1+5) + 1/(1+1)	0.83
doc 5	1/(1+3) + 1/(1+2) + 1/(1+3)	0.83

Example

Position	Document	Score
1	doc 2	1.08
2	doc 3	1.03
3	doc 4	0.83
4	doc 5	0.83
5	doc 1	0.57

"Blended"
not normalized

Advantages

While supervised learning-to-rank methods have garnered much attention of late, unsupervised methods are attractive because they require no training examples.

Advantages

We found that RRF, when used to combine the results of IR methods (including learning to rank), almost invariably improved on the best of the combined results.


This is either madness or brilliance

Risk-Reward Trade-offs in Rank Fusion

https://rodgerbenham.github.io/bc17-adcs.pdf

CombSUM	Fox and Shaw [15]	$\sum_{d \in D} S(d)$
CombMNZ	Fox and Shaw [15]	$\lvert d \in D \rvert \cdot \sum_{d \in D} S(d)$
Borda	de Borda [12]	$\frac{n - r(d) + 1}{n}$
RRF	Cormack et al. [10]	$\sum_{d \in D} \frac{1}{k + r(d)}$
ISR	Mourao et al. [26]	$\lvert d \in D \rvert \cdot \sum_{d \in D} \frac{1}{r(d)^2}$
logISR	Mourao et al. [26]	$\log(\lvert d \in D \rvert) \cdot \sum_{d \in D} \frac{1}{r(d)^2}$
RBC	Bailey et al. [3]	$\sum_{d \in D} (1 - \phi)\phi^{r(d)-1}$

RRF in

Elasticsearch

Weaviate

Azure Cognitive Search

elastic Relevance Workbench

matrix

Movies

Elastic Learned Sparse Encoder

Enable RRF

Score: 13.793848 +1

The Matrix Revisited

The film goes behind the scenes of the 1999 sci-fi movie The Matrix.

Score: 13.608998 -1

Making 'Enter the Matrix'

BM25

Score: 11.082844

Making 'Enter the Matrix'

A look at Enter the Matrix: The game's story picks up just before The Matrix Reloaded and runs parallel to that of the film. Bend the rules of the Matrix with martial arts, pilot the fastest hovercraft in the fleet, or just...

Score: 10.003275

The Matrix Revisited

```
GET quotes/_search
{
    "query": {
        "match": {
            "quote": "pirates"
        }
    },
    "knn": {
        "field": "vector",
        "query_vector": [1.25, 2, 3.5],
        "k": 50,
        "num_candidates": 100
    },
    "rank": {
        "rrf": {
            "window_size": 50,
            "rank_constant": 20
        }
    }
}
```

Full (minimal) example

Including an aggregation

```
PUT example-rrf
{
  "mappings": {
    "properties": {
      "text": {
        "type": "text"
      },
      "vector": {
        "type": "dense_vector",
        "dims": 1,
        "index": true,
        "similarity": "l2_norm"
      },
      "integer": {
        "type": "integer"
      }
    }
  }
}
```

```
PUT example-rrf/_doc/1
{
    "text" : "rrf",
    "vector" : [5],
    "integer": 1
}
PUT example-rrf/_doc/2
{
    "text" : "rrf rrf",
    "vector" : [4],
    "integer": 2
}
PUT example-rrf/_doc/3
{
    "text" : "rrf rrf rrf",
    "vector" : [3],
    "integer": 1
}
PUT example-rrf/_doc/4
{
    "text" : "rrf rrf rrf rrf",
    "integer": 2
}
PUT example-rrf/_doc/5
{
    "vector" : [0],
    "integer": 1
}
```

```
GET example-rrf/_search
{
    "query": {
        "term": {
            "text": "rrf"
        }
    },
    "knn": {
        "field": "vector",
        "query_vector": [3],
        "k": 5,
        "num_candidates": 5
    },
    "rank": {
        "rrf": {
            "window_size": 5,
            "rank_constant": 1
        }
    },
    "size": 3,
    "aggs": {
        "int_count": {
            "terms": {
                "field": "integer"
            }
        }
    }
}
```

```
GET example-rrf/_search
{
  "query": {
    "term": {
      "text": "rrf"
    }
  }
}

GET example-rrf/_search
{
  "knn": {
    "field": "vector",
    "query_vector": [3],
    "k": 5,
    "num_candidates": 5
  }
}
```

Document	Query + kNN	Score
doc 1	1/(1+4) + 1/(1+3)	0.45
doc 2	1/(1+3) + 1/(1+2)	0.58
doc 3	1/(1+2) + 1/(1+1)	0.83
doc 4	1/(1+1)	0.50
doc 5	1/(1+4)	0.20

Settings

k: default 60 (optional)

Window size: default 100 (optional)

Not supported (8.10)

scroll, point in time, sort, rescore, suggesters, highlighting, collapse, explain, profiling


PARLAY

Conclusion

RRF: good results, easy to understand and calculate, no tuning

PS: Rank Quest

https://rankquest.jillesvangurp.com

Reciprocal Rank Fusion

How to Stop Worrying about Boosting

Philipp Krenn @xeraa