ScoresAsPercentages

Scores As Percentages

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.

Here is an elaboration from the mailing list...

http://www.nabble.com/Re%3A-How-to-het-the-score-in-percentage-p23340308.html

```
: here ie, in our existing system we are showing the search score in
: percenetage but lucene provides the search score in numbers which is derived
: from some internal logic. Can anybody give some tips for converting the
: lucene score to percentage or is there any way to retrive the score as
: percentage from lucene search.

there is an extremely important and fundemental question you have to
answer when you say you want "the score as a percentage" ...

        A percentage of what exactly?

As Erick has pointed out: it's easy to convert a document's numeric score
into a percentage of the highest scoring document using division ... but
that is just as meaningless as dividing by pi.  Consider the following
query...

   query = apple eclipse zzz yyy xxx qqq kkk ttt rrr

Now imagine that three "documents" match this query, with the arbitrary
scores listed...

   2.345 doc1: apple bannana
  16.415 doc2: zzz yyy xxx qqq kkk ttt rrr 111
   2.345 doc3: eclipse moon sun

(we're ignoring idf and norms for the moment).  obviously doc2 has the
highest score, and is in fact a *very* good match for your query, so it's
fine that it gets a score of 100%.  doc1 and doc2 each get scores of 14%.

now what happens if doc2 gets deleted from my index?

doc1 and doc3 both still match the query, and they now both tie for the
"highest" score, so now suddenly they have scores of 100%.

This is going to confuse the hell out of your users.  the query hasn't
changed, doc1 and doc2 didn't change.  doc1 didn't suddenly become 7
times more relevant to your query then it was 5 minutes ago.

you might say: "i'm never going to delete documents", but are you ever
going to add documents?  because if so you're going to have the same
problem.  if you never modify your index at all ... well then i envy you,
but you're still going to have a problem -- namely that people are going
to look at a docA that scores 92% against queryX and a differnet docB that
scores 68% against queryY and say "WTF? docB is a much better match queryY
then docA is for query!?!?"

score values are meaningful only for purposes of comparison between other
documents for the exact same query and the exact same index.  when you try
to compute a percentage, you are setting up an implicit comparison with
scores from other queries.
```

In some other threads, the topic of computing a percentage from the maximum *possible* score has been discussed, but an approach like this would pose additional problems...

http://www.nabble.com/theoretical-maximum-score-to17147485.html#a17285093

```
: Is it possible to compute a theoretical maximum score for a given query if
: constraints are placed on 'tf' and 'lengthNorm'? If so, scores could be
: compared to a 'perfect score' (a feature request from our customers)

without thinking about it two hard, you'd also need to constrain:
 * field and document boosts (which are combined with lengthNorm to create
   the fieldNorm)
 * query time boosts
 * idf (there's no law that says Similarity.idf has to return a number
   less then 1)
 * queryNorm

...you'd also need to use query structures that are simple -- a
ValueSourceQuery can break all the rules it wants -- but if you used only
basic types of queries (boolean, term, phrase) and you imposed those
constraints you could probably pull it off.

the key thing to watch out for is that even if you can do it, and you can
start to say meaninful things like "doc A scored X out of a max possible Y
against query Q" that doesn't neccessarily help you compare that with doc
B which scored V out of a max possible W against query R ... even if X/Y == V/W
... because the *structure* and complexity of Q and R play havock
with the scores.

but comparisons like that are what people are going to start to do as
soome as you give them a number like that.  People are going to start to
tihnk "well doc A is an N% match for Q, and doc B is an N% match for R,
but A is clearly a better match Q then B is for R .. what the heck?"

...i would do a lot of "subjective" testing using a variety of queries of
various complexities before i put any faith in producing a number like
that.

I suspect what you'd find is that the constraints you need to impose in
order to give you a meaningful number wind up hobbling Lucene so much it
doesn't do a very good job of scoring anything.
```

As alluded to in that email, this still wouldn't make the scores comparable between queries with different structures; It also would suffer the same problem of percentages changing when documents in the index are added/removed (because of idf changes) even if those documents have nothing to do with the query.

More background...

- http://article.gmane.org/gmane.comp.jakarta.lucene.user/12076
- http://article.gmane.org/gmane.comp.jakarta.lucene.user/10810