

CommunityContributions

Community Contributions

Demoting results

The `BoostingQuery` class can be used to effectively demote results that match a given query. Unlike the "NOT" clause, this still selects documents that contain undesirable terms, but reduces their overall score:

```
Query balancedQuery = new BoostingQuery(positiveQuery, negativeQuery, 0.01f);
```

In this scenario the `positiveQuery` contains the mandatory, desirable criteria which is used to select all matching documents, and the `negativeQuery` contains the undesirable elements which are simply used to lessen the scores. Documents that match the `negativeQuery` have their score multiplied by the supplied "boost" parameter, so this should be less than 1 to achieve a demoting effect.

This code was made available here: [http://marc.theaimsgroup.com/?l=ucene-user&m=108058407130459&w=2](http://marc.theaimsgroup.com/?l=.lucene-user&m=108058407130459&w=2)

Utilities

[LSql](#) is a command-line tool that allows the user to run sql-like queries against the Lucene index. This tool is useful for automated migrations, because it allows the developer to treat a Lucene index the same as you would a database.

[Luke](#) is an absolutely terrific utility for testing your lucene indices and queries dynamically through a GUI.

Extensions

[Zoie](#) is a realtime search system built on Lucene.

[Bobo](#) is a facet search extension to Lucene.