Advanced fulltext search with Sphinx

Adrian Nuta // Sphinxsearch // 2014
Fulltext search in MySQL

- available for MyISAM and lately for InnoDB
- limited in indexation options
  - only min length and list of stopwords
- limited in search options
  - boolean
  - natural mode
  - with query expansion
Why Sphinx?

- GPLv2
- better performance
- lot of features, both on indexing and searching
- easy to transit from MySQL:
  - easy to index from MySQL
  - SphinxQL - access and query Sphinx using any MySQL client
MySQL vs Sphinx fulltext index

- B-tree index
- easy to update frequently, easy to access by PK
- columnar storage
- OLTP

- inverted index
- hard to update, fast to read
- keyword based storage
- OLAP
Simple fulltext search

MySQL:
mysql> SELECT * FROM myindex
       WHERE MATCH('title,content') AGAINST ('find me fast');

Sphinx:
mysql> SELECT * FROM myindex
       WHERE MATCH('find me fast');
More complete Sphinx search

```sql
mysql> SELECT * FROM index WHERE MATCH('"a quorum search is made here"/4')
ORDER BY WEIGHT() DESC, id ASC
OPTION ranker = expr(
  'sum(
    exact_hit+10*(min_hit_pos==1)+lcs*(0.1*my_attr)
  )*1000 +
  bm25'
);
```
Searching only on some fields

- Not possible in MySQL, need to declare separate index
- in Sphinx - syntax operator:

```sql
mysql> SELECT * FROM myindex
    
    WHERE MATCH('@(title,content) find me fast');
```
Indexing features

- charset table
- stopwords, wordforms
- stemming and lemmatization
- HTML stripping
- blending, ignore chars, bigram words
- custom regexp filters
Searching operators

- wildcard
- proximity
- phrase
- start/end
- qourum matching
- strict order
- sentence, paragraph, HTML zone limitation
Ranking factors formulas

- bm25
- LCS - distance between query and document
- word and hit counting
- tf_idf and idf
- word positioning
- possible to use attribute values
### Ranking without field weighting

```sql
mysql> SELECT id, title, weight() FROM wikipedia WHERE MATCH('inverted index') OPTION ranker=expr('sum(hit_count*user_weight)'), field_weights=(title=1, body=1);
```

<table>
<thead>
<tr>
<th>id</th>
<th>title</th>
<th>weight()</th>
</tr>
</thead>
<tbody>
<tr>
<td>221501516</td>
<td>Index (search engine)</td>
<td>125</td>
</tr>
<tr>
<td>221487412</td>
<td>Inverted index</td>
<td>47</td>
</tr>
</tbody>
</table>

Doc. 221501516: 1 hit in ‘title’ x 100 + 124 hits in ‘body’ = **125**

Doc. 221487412: 2 hits in ‘title’ x 100 + 45 hits in ‘body’ = **47**
Ranking with field weighting

mysql> SELECT id, title, WEIGHT() FROM index WHERE MATCH('inverted index') OPTION ranker=expr('sum(hit_count*user_weight)'), field_weights=(title=100, body=1);

+-------------+----------------------------------+-------------+
| id          | title                            | WEIGHT()    |
|-------------+----------------------------------+-------------+
| 221487412   | Inverted index                   | 245         |
| 221501516   | Index (search engine)            | 224         |

Doc. 221501516: 1 hit in ‘title’ x 100 + 124 hits in ‘body’ = 100+124 = 224
Doc. 221487412: 2 hits in ‘title’ x 100 + 45 hits in ‘body’ = 200+45 = 245
### Words proximity

```sql
mysql> SELECT id, title, WEIGHT() FROM index
WHERE MATCH('@title list of football players') OPTION ranker=expr('sum(lcs)');
```

<table>
<thead>
<tr>
<th>id</th>
<th>title</th>
<th>weight()</th>
</tr>
</thead>
<tbody>
<tr>
<td>207381464</td>
<td>List of football players from Amsterdam</td>
<td>4</td>
</tr>
<tr>
<td>221196229</td>
<td>List of Football Kingz F.C. players</td>
<td>3</td>
</tr>
<tr>
<td>210456301</td>
<td>List of Florida State University football players</td>
<td>2</td>
</tr>
</tbody>
</table>
word and hit count

mysql> SELECT id, title, WEIGHT() AS w FROM index WHERE MATCH('@title php | api') OPTION ranker=expr('sum(hit_count)');
+--------+----------------------------------------------------------+------+
| id     | title                                                   | w    |
+--------+----------------------------------------------------------+------+
| 1000671 | PHP API gives PHP Warnings - tips?                      | 3    |
...

mysql> SELECT id, title, WEIGHT() AS w FROM index WHERE MATCH('@title php | api') OPTION ranker=expr('sum(word_count)');
+--------+----------------------------------------------------------+------+
| id     | title                                                   | w    |
+--------+----------------------------------------------------------+------+
| 1000671 | PHP API gives PHP Warnings - tips?                      | 2    |
Position

mysql> `select id, title, weight() as w from forum where match('@title sphinx php api') option ranker=expr('sum(min_hit_pos)');`

<table>
<thead>
<tr>
<th>id</th>
<th>title</th>
<th>w</th>
</tr>
</thead>
<tbody>
<tr>
<td>1004955</td>
<td>how can i do a sample search use sphinx php api</td>
<td>9</td>
</tr>
<tr>
<td>1004900</td>
<td>How to update fulltext field using sphinx api of PHP?</td>
<td>7</td>
</tr>
<tr>
<td>1008783</td>
<td>Update MVA-Attributes with the PHP-API Sphinx 2.0.2</td>
<td>6</td>
</tr>
<tr>
<td>1000498</td>
<td>Limits in sphinx when using PHP sphinx API</td>
<td>3</td>
</tr>
</tbody>
</table>

how can i do a sample search use sphinx php api

```sql
1 2 3 4 5 6 7 8 9
```
mysql> select id, title, weight() from wikipedia where match('@title (Polyphonic | Polysyllabic | Oberheim) ') option ranker=expr('sum(max_idf)*1000');

+------------+---------------------------+----------+
| id         | title                     | weight() |
|------------+---------------------------+----------+
| 165867281  | The Polysyllabic Spree    | 112      | Polysyllabic - rare
| 208650218  | Oberheim Xpander          | 108      | Oberheim - not so rare
| 209138112  | Oberheim OB-8             | 108      |
| 180503990  | Polyphonic Era            | 85       | Polyphonic - common
| 183135294  | Polyphonic C sharp        | 85       |
| 219939232  | Polyphonic HMI            | 85       |
+------------+---------------------------+----------+
BM25F

mysql> select ... where match('odbc') option ranker=expr('1000*bm25f(1,1)');

+---+-------------------------+--------+--------+-----------+
<table>
<thead>
<tr>
<th>id</th>
<th>title</th>
<th>title_len</th>
<th>body_len</th>
<th>weight()</th>
</tr>
</thead>
<tbody>
<tr>
<td>179</td>
<td>odbc_dsn</td>
<td>1</td>
<td>69</td>
<td>775</td>
</tr>
<tr>
<td>170</td>
<td>type</td>
<td>1</td>
<td>124</td>
<td>742</td>
</tr>
</tbody>
</table>
...

mysql> select ... where match('odbc') option ranker=expr('1000*bm25f(1,0)');

+---+------------------------+--------+--------+-----------+
<table>
<thead>
<tr>
<th>id</th>
<th>title</th>
<th>title_len</th>
<th>body_len</th>
<th>weight()</th>
</tr>
</thead>
<tbody>
<tr>
<td>169</td>
<td>Data source configuration options</td>
<td>4</td>
<td>6246</td>
<td>758</td>
</tr>
<tr>
<td>179</td>
<td>odbc_dsn</td>
<td>1</td>
<td>69</td>
<td>743</td>
</tr>
<tr>
<td>170</td>
<td>type</td>
<td>1</td>
<td>124</td>
<td>689</td>
</tr>
</tbody>
</table>
Language morphology

Will the user search ‘shirt’ or ‘shirts’?

- stemming:
  - shirt = shirts

- index_exact_form for exact matching

- lemmatization:
  - men = man
EF-S 18-200mm f/3.5-5.6

blend_chars

- act as both separators and valid chars
- **10-200mm** with - blended will index 3 terms: 10-200mm, 10 and 200mm
- leading or trailing blend char behaviour can be configured to be stripped or indexed
Sentence delimitation

```sql
mysql> INSERT INTO index VALUES(1, 'quick brown fox jumps over the lazy dog');
mysql> INSERT INTO index VALUES(2, 'The quick brown fox made it. Where was the lazy dog?');
mysql> SELECT * FROM index WHERE MATCH('brown fox SENTENCE lazy dog');
```

```
+----+
| id |
+----+
| 1  |
+----+
```
mysql> INSERT INTO index VALUES(1, 'The quick brown fox jumps over the lazy dog'');
mysql> INSERT INTO index VALUES(2, 'The quick brown fox jumps over the lazy dog');

mysql> SELECT * FROM index WHERE MATCH('brown fox PARAGRAPh lazy dog');

+-----+
| id  |
+-----+
| 1   |
+-----+
More fulltext features

- bigrams
- more ranking factors: lccs, wlccs, atc
- phrase boundary chars
- HTML index attributes, elements removal
- RLP Chinese tokenization
- position step tuning
Thank you!

http://www.sphinxsearch.com
adrian.nuta@sphinxsearch.com