黑客派

深入 Fuzzy Query

作者：felayman
原文链接：https://hacpai.com/article/1512989028502
来源网站：黑客派

```
<h2 id="Fuzzy-Query">Fuzzy Query</h2>
<script async src="https://pagead2.googlesyndication.com/pagead/js/adsbygoogle.js"></scr
pt>
<!-- 黑客派PC帖子内嵌-展示 -->
<ins class="adsbygoogle" style="display:block" data-ad-client="ca-pub-5357405790190342"
data-ad-slot="8316640078" data-ad-format="auto" data-full-width-responsive="true"></in
>
<script>
    (adsbygoogle = window.adsbygoogle || []).push({});
</script>
<blockquote>
 <p>国内对于 Elasticsearch 深入的人并不多，或者大多数大牛不屑于分享关于 Elasticsearch 的知
，这里讲讲 Elasticsearch 中的 Fuzzy Query</p>
</blockquote>
<h3 id="关于">关于</h3>
<ul>
 <li>官方文档：<a href="https://link.hacpai.com/forward?goto=https%3A%2F%2Fwww.elastic
co%2Fguide%2Fen%2Felasticsearch%2Freference%2Fcurrent%2Fquery-dsl-fuzzy-query.html"
arget="_blank" rel="nofollow ugc">官方文档</a></li>
 <li>ApacheCN : <a href="https://link.hacpai.com/forward?goto=http%3A%2F%2Fcwiki.apac
ecn.org%2Fpages%2Fviewpage.action%3FpageId%3D4882439" target="_blank" rel="nofollow
ugc">中文文档</a></li>
</ul>
<h3 id="概念">概念</h3>
<pre><code class="highlight-chroma">模糊查询利用了基于Levenshtein编辑距离的相似度
关于什么是编辑距离(Levenshtein)以及算法,请查看下面的参考部分.
</code></pre>
<h3 id="语法">语法</h3>
<pre><code class="highlight-chroma">GET /_search
{
   "query": {
      "fuzzy" : {
         "user" : {
            "value" :        "ki",
               "boost" :        1.0,
               "fuzziness" :     2,
               "prefix_length" : 0,
               "max_expansions": 100
         }
      }
   }
}
</code></pre>
<p><strong>参数说明</strong></p>
<ul>
 <li>fuzziness 控制编辑距离(目前只支持 0,1,2)</li>
 <li>boost 设置查询权重</li>
 <li>prefix_length 设置匹配的 term 的前 prefix_length 个字符不会参与模糊查询</li>
 <li>max_expansions 控制最大的返回结果</li>
</ul>
<h3 id="Fuzziness---模糊性--">Fuzziness ( 模糊性 )</h3>
<p>当查询 text ( 文本 ) 或者 keyword fields ( 关键字字段 )时，模糊性被解释为 Levenshtein Edit D
stance —— 是指两个字串之间，由一个转成另一个所需的最少编辑操作次数 max_expansions 来控
查询结果。</p>
```

模糊性参数可以指定为：
0， 1， 2
最大允许 Levenshtein Edit Distance （或者编辑次数）。
AUTO
基于该项的长度 generates an edit distance (生成编辑距离)。对于长度：
0..2
 必须完全匹配
 3..5
 允许 one edi
allowed (编辑一次)
 >5
 允许 two edits allowed (编辑两次)<
p>
<blockquote>
<img src="https://static.hacpai.com/images/img-loading.svg" alt="import" data-src="ht
ps://www.elastic.co/guide/en/elasticsearch/reference/5.5/images/icons/tip.png"> 目前 Elastic
earch 仅支持编辑距离 =2 的查询，因为该操作比较重，在使用的时候，请最好使用
</blockquote>
<h3 id="Java-API">Java API</h3>
<pre><code class="language-java highlight-chroma">/**
  * fuzzyQuery基于编辑距离(Levenshtein)来进行相
搜索,比如搜索kimzhy,可以搜索出kinzhy(编辑距离为1)
  * 为了进行测试说明,前创建一个索引,插入几条数据ka,
ab,kib,ba,我们的搜索源为ki
  * 了解更多关于编辑距离(Levenshtein)的概念,请参考:
lt;a href='http://www.cnblogs.com/biyeymyhjob/archive/2012/09/28/2707343.html'></
>
  * 了解更多编辑距离的算法,请参考:<a href='http://
log.csdn.net/ironrabbit/article/details/18736185'>
  * ki — ka 编辑距离为1
  * ki — kab 编辑距离为2
  * ki — kbia 编辑距离为3
  * ki — kib 编辑距离为1
  * 所以当我们设置编辑距离(ES中使用fuzziness参数
控制)为0的时候,没有结果
  * 所以当我们设置编辑距离(ES中使用fuzziness参数
控制)为1的时候,会出现结果ka,kib
  * 所以当我们设置编辑距离(ES中使用fuzziness参数
控制)为2的时候,会出现结果ka,kib,kab
  * 所以当我们设置编辑距离(ES中使用fuzziness参数
控制)为3的时候,会出现结果ka,kib,kab,kbaa(很遗憾,ES本身最多只支持到2,因此不会出现此结果)
  */
    QueryBuilder qb
= QueryBuilders<sp
n class="highlight-o">.fuzzyQuery<span class=
highlight-o">("username"<span class="highlight
o">,"ki")
//   .fuzziness(Fuzziness.ZERO); 没有结果
//   .fuzziness(Fuzziness.ONE); 会出现结果ka,kib
//   .fuzziness(Fuzziness.TWO);会出现结果ka,kib,k
b
            .
span class="highlight-na">fuzziness(<span clas
="highlight-n">Fuzziness.<span class="highligh
-na">AUTO); //会
现结果ka,kib,kab
      SearchRespon
e response =</span
 client.<span class="

ighlight-na">prepareSearch()
.setIndices</span
("index"<span class=
highlight-o">)
.setTypes
span class="highlight-o">("type"<span class="h
ghlight-o">)
.setQuery</span
(qb<span class="hig
light-o">)
.execute<
pan class="highlight-o">()
.actionGet</span
();
</code></pre>
<script async src="https://pagead2.googlesyndication.com/pagead/js/adsbygoogle.js"></scr
pt>
<!-- 黑客派PC帖子内嵌-展示 -->
<ins class="adsbygoogle" style="display:block" data-ad-client="ca-pub-5357405790190342"
data-ad-slot="8316640078" data-ad-format="auto" data-full-width-responsive="true"></in
>
<script>
(adsbygoogle = window.adsbygoogle || []).push({});
</script>
代码详解请参考：<a href="https://link.hacpai.com/forward?goto=https%3A%2F%2Fgithu
.com%2Ffelayman%2Felasticsearch-java-api%2Fblob%2Fmaster%2Fsrc%2Ftest%2Fjava%2For
%2Fvisualchina%2Felasticsearch%2Fapi%2Fdemo%2Fquery%2FFuzzyQueryDemo.java" target
"_blank" rel="nofollow ugc">FuzzyQueryDemo.java
<h2 id="参考">参考</h2>

<a href="https://link.hacpai.com/forward?goto=https%3A%2F%2Fen.wikipedia.org%2Fw
ki%2FDamerau%25E2%2580%2593Levenshtein_distance" target="_blank" rel="nofollow ugc"
Damerau–Levenshtein distance
<a href="https://link.hacpai.com/forward?goto=http%3A%2F%2Fwww.cnblogs.com%2Fb
yeymyhjob%2Farchive%2F2012%2F09%2F28%2F2707343.html" target="_blank" rel="nofollow
ugc">编辑距离及编辑距离算法
<a href="https://link.hacpai.com/forward?goto=http%3A%2F%2Fblog.csdn.net%2Fironr
bbit%2Farticle%2Fdetails%2F18736185" target="_blank" rel="nofollow ugc">Java 算法之 Lev
nshtein Distance（编辑距离）算法
<a href="https://link.hacpai.com/forward?goto=http%3A%2F%2Fcwiki.apachecn.org%2
pages%2Fviewpage.action%3FpageId%3D4882851" target="_blank" rel="nofollow ugc">常
选项
