

## 使用查询重写字段值（敏感词搜索）

这个功能可以解决敏感词搜索问题。

场景：如果创建一个与Mapping字段相同名称的运行时字段名，那么这个运行时字段会覆盖Mapping字段，并按照你定义的脚本返回这个字段值。

要求：敏感词不能展示，但是可以被搜索。

如，定义敏感词为：“敏感”，此时搜索“敏感”二字可以将这个文本搜索出来，原始文本不变（毕竟敏感词多了全是\*已经认不出原文了），但不能将敏感两个字在搜索结果中展示出来。

这条数据 {"text": "我是一个敏感的内容"} 搜索关键词“敏感”时，这条数据应展示为“我是一个\*\*的内容”。

```
POST my_d_mapping_002/_bulk?refresh=true
{"index":{}}
{"text": "我是一个可以正常返回的内容"}
{"index":{}}
{"text": "我是一个可以正常返回的内容"}
{"index":{}}
{"text": "我是一个可以正常返回的内容"}
{"index":{}}
{"text": "我是一个可以正常返回的内容"}
{"index":{}}
{"text": "我是一个可以正常返回的内容"}
{"index":{}}
{"text": "我是一个可以正常返回的内容"}
{"index":{}}
{"text": "我是一个敏感的内容"} //搜索关键词“敏感”时，这条数据应展示为我是一个**的内容。
```

使用运行时字段在查询时覆盖掉原始原始字段text

```
POST my_d_mapping_002/_search
{
  "runtime_mappings": {
    "text": { //运行时字段与Mapping中需要被覆盖的字段名称相同。
      "type": "keyword",
      "script": {
        "source": //如果text文本中有敏感两个字
        """
emit(/[敏感]/.matcher(doc['text.keyword'].value).replaceAll('*')) //表示如果
text.keyword中包含“敏感”一词时，将“敏感”词变为”*“
        """
      }
    }
  },
  "query": {
    "query_string": {
      "default_field": "text.keyword",
      "query": "*敏感*"
    }
  },
  "fields": [
    "text"
  ]
}
```

```
]
}
```

结果可以看到这条数据被查询出来，且fields里的text已经脱敏展示。原本的\_source内容没有改变。

```
5 {"text": "我是一个可以正常返回的内容"}
6 {"index": {}}
7 {"text": "我是一个可以正常返回的内容"}
8 {"index": {}}
9 {"text": "我是一个可以正常返回的内容"}
10 {"index": {}}
11 {"text": "我是一个可以正常返回的内容"}
12 {"index": {}}
13 {"text": "我是一个可以正常返回的内容"}
14 {"index": {}}
15 {"text": "我是一个可以正常返回的内容"}
16 {"index": {}}
17 {"text": "我是一个敏感的内容"}
18
19 GET my_d_mapping_002/_mapping
20
21 GET my_d_mapping_002/_search
22 {
23   "query": {
24     "match": {
25       "text": "敏感"
26     }
27   }
28 }
29
30 POST my_d_mapping_002/_search
31 {
32   "runtime_mappings": {
33     "text": {
34       "type": "keyword",
35       "script": {
36         "source": "//如果文本中有敏感两个字
37         emit([[敏感]].matcher(doc['text.keyword'].value).replaceAll('*'))
38       }
39     }
40   }
41 }
42
43 {
44   "query": {
45     "query_string": {
46       "default_field": "text.keyword",
47       "query": "*敏感*"
48     }
49   },
50   "fields": [
51     "text"
52   ]
53 }
54
```

```
1 {
2   "took": 7,
3   "timed_out": false,
4   "_shards": {
5     "total": 1,
6     "successful": 1,
7     "skipped": 0,
8     "failed": 0
9   },
10  "hits": {
11    "total": {
12      "value": 1,
13      "relation": "eq"
14    },
15    "max_score": 1,
16    "hits": [
17      {
18        "_index": "my_d_mapping_002",
19        "_id": "cuULEpEBk-wCeltBa0W",
20        "_score": 1,
21        "_source": {
22          "text": "我是一个敏感的内容"
23        },
24        "fields": {
25          "text": {
26            "value": "我是一个**的内容"
27          }
28        }
29      }
30    ]
31  }
32 }
```

如果需要原始source不展示，在Mapping中这时`source`参数为false

//注意，需要在创建索引时设置。

```
PUT my_d_mapping_002
{
  "mappings": {
    "_source": {
      "enabled": false
    }
  }
}
```

```
1 {"text": "我是一个敏感的内容"}
2
3 GET my_d_mapping_002/_mapping
4
5 PUT my_d_mapping_002
6 {
7   "mappings": {
8     "source": {
9       "enabled": false
10    }
11  }
12 }
13
14 GET my_d_mapping_002/_search
15 {
16   "match": {
17     "text": "敏感"
18   }
19 }
20
21 POST my_d_mapping_002/_search
22 {
23   "runtime_mappings": {
24     "text": {
25       "type": "keyword",
26       "script": {
27         "source": "//如果文本中有敏感两个字
28         emit([[敏感]].matcher(doc['text.keyword'].value).replaceAll('*'))
29       }
30     }
31   }
32   "query": {
33     "query_string": {
34       "default_field": "text.keyword",
35       "query": "*敏感*"
36     }
37   },
38   "fields": [
39     "text"
40   ]
41 }
42
```

```
1 {
2   "took": 5,
3   "timed_out": false,
4   "_shards": {
5     "total": 1,
6     "successful": 1,
7     "skipped": 0,
8     "failed": 0
9   },
10  "hits": {
11    "total": {
12      "value": 1,
13      "relation": "eq"
14    },
15    "max_score": 1,
16    "hits": [
17      {
18        "_index": "my_d_mapping_002",
19        "_id": "eelVpEBk-wCeltBFEXC",
20        "_score": 1,
21        "fields": {
22          "text": {
23            "value": "我是一个**的内容"
24          }
25        }
26      }
27    ]
28  }
29 }
```

## 对比\_update\_by\_query

脱敏处理还有一个API看起来好像也能做到那就是update\_by\_query。

这个API，顾名思义在查询时更新。那一起来看看这个区别。

```
PUT my_d_mapping_003/_bulk?refresh
{"index":{}}
{"text":"我是一个可以正常返回的内容"}
{"index":{}}
{"text":"我是一个可以正常返回的内容"}
{"index":{}}
{"text":"我是一个可以正常返回的内容"}
{"index":{}}
{"text":"我是一个可以正常返回的内容"}
{"index":{}}
{"text":"我是一个可以正常返回的内容"}
{"index":{}}
{"text":"我是一个可以正常返回的内容"}
{"index":{}}
{"text":"我是一个敏感的内容"}
```

调用\_update\_by\_query,jiang 字段text进行敏感字替换。同样的这条数据 {"text":"我是一个敏感的内容"} 搜索关键词“敏感”时，这条数据应展示为“我是一个\*\*的内容”。

```
POST my_d_mapping_003/_update_by_query
{
  "script": {
    "lang": "painless",
    "source": "ctx._source.text = /[敏感]/.matcher(ctx._source.text).replaceAll('*')"
  }
}
```

执行查询语句

```
GET my_d_mapping_003/_search
{
  "query": {
    "query_string": {
      "default_field": "text.keyword",
      "query": "****"
    }
  }
}
```

```
DELETE my_d_mapping_003
PUT my_d_mapping_003/_bulk?refresh
{"index":{}}
{"text":"我是一个可以正常返回的内容"}
{"index":{}}
{"text":"我是一个可以正常返回的内容"}
{"index":{}}
{"text":"我是一个可以正常返回的内容"}
{"index":{}}
{"text":"我是一个可以正常返回的内容"}
{"index":{}}
{"text":"我是一个可以正常返回的内容"}
{"index":{}}
{"text":"我是一个敏感的内容"}
{"index":{}}
{"text":"我是一个敏感的内容"}

POST my_d_mapping_003/_update_by_query
{
  "script": {
    "lang": "painless",
    "source": "ctx._source.text = /[敏感]/.matcher(ctx._source.text).replaceAll('*')"
  }
}

GET my_d_mapping_003/_search
{
  "query": {
    "query_string": {
      "default_field": "text.keyword",
      "query": "****"
    }
  }
}

GET my_d_mapping_003/_search
{
  "runtime_mappings": {
    "last": {
      "type": "keyword",
      "script": {
        "source": "/*如果text文本中有敏感两个字*/"
      }
    }
  },
  "query": {
    "query_string": {
      "default_field": "text.keyword",
      "query": "****"
    }
  }
}
```

看起来数据被查出来了还脱敏了。

实际上仔细想想我们之前的业务需求，除了需要脱敏，还需要保证原始文本内容不变，那么就看一下原始文本内容。

```
34 {"text":"我是一个可以正常返回的内容"}
35 {"index":{}}
36 {"text":"我是一个可以正常返回的内容"}
37 {"index":{}}
38 {"text":"我是一个敏感的内容"}
39 {"index":{}}
40 {"text":"我是一个敏感的内容"}

41 POST my_d_mapping_003/_update_by_query
42 {
43   "script": {
44     "lang": "painless",
45     "source": "ctx._source.text = /[敏感]/.matcher(ctx._source.text).replaceAll('*')"
46   }
47 }
48
49 GET my_d_mapping_003/_search
50 {
51   "runtime_mappings": {
52     "last": {
53       "type": "keyword",
54       "script": {
55         "source": "/*如果text文本中有敏感两个字*/"
56       }
57     }
58   },
59   "query": {
60     "query_string": {
61       "default_field": "text.keyword",
62       "query": "****"
63     }
64   }
65 }

66 GET my_d_mapping_003/_search
67 {
68   "runtime_mappings": {
69     "last": {
70       "type": "keyword",
71       "script": {
72         "source": "/*如果text文本中有敏感两个字*/"
73       }
74     }
75   },
76   "query": {
77     "query_string": {
78       "default_field": "text.keyword",
79       "query": "****"
80     }
81   }
82 }
```

原始文本已经被替换成了\*\*。细心的人已经发现了，我第一次搜索的不是“脱敏”，而是“\*\*”，当我搜索“脱敏”一词是无法将文档搜索出来的。

```
61 GET my_d_mapping_003/_search
62 {
63 }
64
65 DELETE my_d_mapping_003
66 PUT my_d_mapping_003/_bulk?refresh
67 {"index":{}}
68 {"text":"我是一个可以正常返回的内容"}
69 {"index":{}}
70 {"text":"我是一个可以正常返回的内容"}
71 {"index":{}}
72 {"text":"我是一个可以正常返回的内容"}
73 {"index":{}}
74 {"text":"我是一个可以正常返回的内容"}
75 {"index":{}}
76 {"text":"我是一个敏感的内容"}
77 {"index":{}}
78 {"text":"我是一个敏感的内容"}
79 {"index":{}}
80 {"text":"我是一个敏感的内容"}
81 {"index":{}}
82 {"text":"我是一个敏感的内容"}
83 {"index":{}}
84 {"text":"我是一个敏感的内容"}
85 {"index":{}}
86 {"text":"我是一个敏感的内容"}
87 {"index":{}}
88 {"text":"我是一个敏感的内容"}
89 {"index":{}}
90 {"text":"我是一个敏感的内容"}
91 {"index":{}}
92 {"text":"我是一个敏感的内容"}
93 {"index":{}}
94 {"text":"我是一个敏感的内容"}
95 {"index":{}}
96 {"text":"我是一个敏感的内容"}
97 {"index":{}}
98 {"text":"我是一个敏感的内容"}
99 {"index":{}}
100 {"text":"我是一个敏感的内容"}

101 POST my_d_mapping_003/_update_by_query
102 {
103   "script": {
104     "lang": "painless",
105     "source": "ctx._source.text = /[敏感]/.matcher(ctx._source.text).replaceAll('*')"
106   }
107 }
108
109 GET my_d_mapping_003/_search
110 {
111   "query": {
112     "query_string": {
113       "default_field": "text.keyword",
114       "query": "****"
115     }
116   }
117 }
118
119 GET my_d_mapping_003/_search
120 {
121   "query": {
122     "query_string": {
123       "default_field": "text.keyword",
124       "query": "****"
125     }
126   }
127 }
```

所以可以看到，\_update\_by\_query这个API实际上是在搜索时将我们的文档进行了替换并查询出来，而使用运行时字段对原Mapping进行替换并不会改变原始的文档内容，还可以根据自己的需要判断是否返回原始内容。

## Painless 语法继续扩展

通过刚才查询出来的内容可以看到这条数据 {"text":"我是一个敏感的内容"} 搜索关键词“敏感”时，这条数据应展示为“我是一个\*\*的内容”。有两个\*。说明语法将“敏感”一词中的两个字分别进行了替换，但很多时候敏感的不是词是一个词或者一句话。所以我们需要脱敏后展示“我是一个\*的内容”

最有可能对结果产生影响的就是脚本里的内容了，让我们看脚本里都有哪些内容。

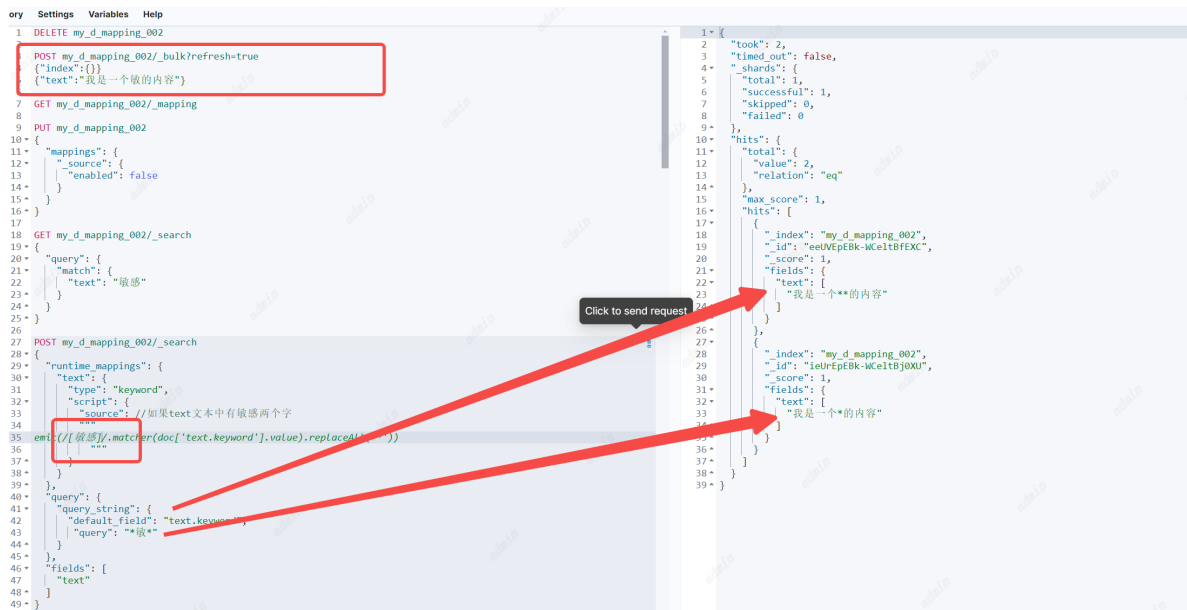
```
emit(/[敏感]/.matcher(doc['text.keyword'].value).replaceAll('*'))
```

1. emit(): 函数处理得到的返回值会赋值给我们定义好的运行时字段
2. /pattern/: 是一个匹配的模型，在painless语法中使用双反斜杠表示一个模型，这里也支持java中的正则表达式。
3. matcher(): 匹配函数
4. doc['text.keyword'].value: 表示获取文档中这个字段的值，这里为什么用keyword呢，因为运行时字段支持的文本类型只有keyword。
5. replaceAll(): 替换函数。

看了这些可以看到问题出现在匹配上，/[敏感]/.matcher 将文本匹配成两个字分别进行后续的替换操作。那为什么不是词呢，关键就是 [] 这个中括号，敏感的程序员一下子就能想到，这表示数组。

没错，这里/[敏感]/在Painless语法中被认为是由 敏 和 感 两个字组成的数组，所以替换后会有两个\*  
可以验证一下，这里我再加入一个数据，这时我搜索敏，可以看到新的数据也被替换了

```
POST my_d_mapping_002/_bulk?refresh=true
{"index":{}}
{"text":"我是一个敏的内容"}
```



那么如何修改呢，当然就是去掉中括号了

```
POST my_d_mapping_002/_search
{
  "runtime_mappings": {
    "text": {
      "type": "keyword",
      "script": {
        "source": "//如果text文本中有敏感两个字
        ""
      }
    }
  }
}
```

```

"query": {
  "query_string": {
    "default_field": "text.keyword",
    "query": "*敏*"
  }
},
"fields": [
  "text"
]
}

```

再次搜索，新加的数据并没有被脱敏。

```

PUT my_d_mapping_002
{
  "mappings": {
    "source": {
      "enabled": false
    }
  }
}

GET my_d_mapping_002/_search
{
  "query": {
    "match": {
      "text": "敏感"
    }
  }
}

POST my_d_mapping_002/_search
{
  "runtime_mappings": {
    "text": {
      "type": "keyword",
      "script": {
        "source": "//如果text文本中有敏感两个字\nemit(/敏感/matcher(doc['text.keyword'].value).replaceAll('*'))\n"}
      }
    },
    "query": {
      "query_string": {
        "default_field": "text.keyword",
        "query": "*敏*"
      }
    },
    "fields": [
      "text"
    ]
  }
}

```

```

{
  "took": 1,
  "timed_out": false,
  "shards": {
    "total": 1,
    "successful": 1,
    "skipped": 0,
    "failed": 0
  },
  "hits": {
    "total": {
      "value": 2,
      "relation": "eq"
    },
    "max_score": 1,
    "hits": [
      {
        "_index": "my_d_mapping_002",
        "_id": "eetUepEBk-WCeltBFEXC",
        "_score": 1,
        "fields": {
          "text": [
            "我是一个*的内容"
          ]
        }
      },
      {
        "_index": "my_d_mapping_002",
        "_id": "ietUepEBk-WCeltBFEXC",
        "_score": 1,
        "fields": {
          "text": [
            "我是一个敏的内容"
          ]
        }
      }
    ]
  }
}

```

```

PUT my_d_mapping_002
{
  "mappings": {
    "source": {
      "enabled": false
    }
  }
}

GET my_d_mapping_002/_search
{
  "query": {
    "match": {
      "text": "敏感"
    }
  }
}

POST my_d_mapping_002/_search
{
  "runtime_mappings": {
    "text": {
      "type": "keyword",
      "script": {
        "source": "//如果text文本中有敏感两个字\nemit(/敏感/matcher(doc['text.keyword'].value).replaceAll('*'))\n"}
      }
    },
    "query": {
      "query_string": {
        "default_field": "text.keyword",
        "query": "*敏*"
      }
    },
    "fields": [
      "text"
    ]
  }
}

```

```

{
  "took": 2,
  "timed_out": false,
  "shards": {
    "total": 1,
    "successful": 1,
    "skipped": 0,
    "failed": 0
  },
  "hits": {
    "total": {
      "value": 1,
      "relation": "eq"
    },
    "max_score": 1,
    "hits": [
      {
        "_index": "my_d_mapping_002",
        "_id": "eetUepEBk-WCeltBFEXC",
        "_score": 1,
        "fields": {
          "text": [
            "我是一个*的内容"
          ]
        }
      }
    ]
  }
}

```

问题：如果有多敏感词呢？