

Please visit website: <http://cxyroad.com>

[tps://p9-juejin.byteimg.com/tos-cn-i-k3u1fbpfcp/b26e6605909c4ae6aeb939dedb13e20c~tplv-k3u1fbpfcp-jj-mark:3024:0:0:0:q75.awebp#?w=487&h=84&s=7004&e=png&b=010101](https://p9-juejin.byteimg.com/tos-cn-i-k3u1fbpfcp/b26e6605909c4ae6aeb939dedb13e20c~tplv-k3u1fbpfcp-jj-mark:3024:0:0:0:q75.awebp#?w=487&h=84&s=7004&e=png&b=010101)

\* 优点：实现简单，同时redis可以提供较高的性能

\* 缺点：

1. 没有利用到业务数据库中的词频等数据
2. 算法太过于简单粗暴，按道理可以通过树或者字符串匹配算法来提高查找效率和存储效率
3. 自动补齐可能补齐不相关的数据，需要特殊处理
4. 不支持热点数据，需要另外一个zset进行记录，然后对取出来的数据进行取交集
5. 不支持纠错功能（因为不是存储的拼音）

## 3.2 用elasticsearch实现

-----

Suggesters API 来实现这个功能

1. Term Suggester（纠错补全，输入错误的情况下补全正确的单词）

1. 这个建议是基于编辑距离，基于已经在数据库中的数据进行分词，然后分割出来术语

2. 所有出来的结果有两种排序方式

1. 第一种（score）：首先按照分数，然后是文档出现的频率，然后是被搜索词本身

2. 第二种（frequency）：首先按照文档出现的频率，然后是分数，然后是术语本身

2. \*\*Phrase Suggester\*\*（自动补全短语，输入一个单词补全整个短语）

1. 这个建议是基于编辑距离，基于已经在数据库中的数据进行分词，然后分割出来术语，但是会根据字符串之间的距离再次提示短语

2. 同时具有纠错功能

3. \\*\*Completion Suggester(\\*\*完成补全单词，输出如前半部分，补全整个单词)

4. \*\*Context Suggester（上下文补全，可以基于Completion Suggester的基础上做）\*\*

1. 可以在插入数据的时候，数据的字段打上特定的标签

2. 然后做sugget搜索的时候，加一个搜索条件，这样的话，就会从特点的数据中进行搜索

3. 标签支持两种：category 和 geo

1. category：就一个或者多个字符串进行标记，比如对于食物，我们可以给一个类型为category 的标签，字段名是place\\_type，然后字段值是

restaurant, 如果命中了对应的一个或者多个place\\_type, 则可以提升分数  
2. geo: 地理上下文可以在索引时将一个或多个地理点与文档字段相关联。在查询时, 如果建议在特定地理位置的一定距离内, 则可以过滤和提升分数

\* \*\*我们前期选择 Completion Suggester 因为这个性能最好, 他将数据保存在内存中的有限状态转移机中 (FST) \*\*  
+ 后期如果需要优化, 可以结合 Term Suggester , Phrase Suggester, Context Suggester进行优化

### ### 3.2.1 建立对应的index (completion 实现)

```
...  
curl -XPUT "localhost:9200/test_index?pretty=true" -H 'content-type:  
application/json' --data '  
{  
  "settings": {  
    "number_of_shards":  
      "_id": "eeZ-bX8BVW9cjfXV7UHR",  
      "_score": 1.0,  
      "_source": {  
        "id": 1,  
        "title": "春节碰到自己的老师怎么办? ",  
        "content": "如题",  
        "status": "1"  
      }  
    },  
    {  
      "text": "春节碰到自己的老师怎么办? 我应该打招呼吗? ",  
      "_index": "test_index",  
      "_type": "_doc",  
      "_id": "euZ-bX8BVW9cjfXV7kEg",  
      "_score": 1.0,  
      "_source": {  
        "id": 1,  
        "title": "春节碰到自己的老师怎么办? 我应该打招呼吗? ",  
        "content": "如题",  
        "status": "1"  
      }  
    }  
  ]  
}  
]
```

...

### ### 3.3.2 match phrase prefix query 实现

![image (10).png](https://p9-juejin.byteimg.com/tos-cn-i-k3u1fbpfcp/89a2bb0890114238bbf8da18e54f7484~tplv-k3u1fbpfcp-jj-mark:3024:0:0:0:q75.awebp#?w=851&h=566&s=81850&e=png&b=fbfbfb)

## 4. 参考链接

=====

### \* 自动提示API集合

: [www.iaspnetcore.com/blog/blogpo...](http://cxyroad.com/"https://www.iaspnetcore.com/blog/blogpost/5a8efc68f5eba4276034179d")

\* 自动补齐: [www.6aiq.com/article/159...](http://cxyroad.com/"https://www.6aiq.com/article/1599611620992")

### \* elasticsearch 7.1 官方支持搜索文档

: [www.elastic.co/guide/en/el...](http://cxyroad.com/"https://www.elastic.co/guide/en/elasticsearch/reference/7.17/search-suggesters.html")

\* [www.cnblogs.com/wangzhuxing...](http://cxyroad.com/"https://www.cnblogs.com/wangzhuxing/p/9574630.html")

原文链接: <https://juejin.cn/post/7363200136976171049>