

基于Elasticsearch实现搜索建议

📅 2017-01-23 | 📁 [Elasticsearch](#) | 📖 阅读次数 1643

搜索建议是搜索的一个重要组成部分，一个搜索建议的实现通常需要考虑建议词的来源、匹配、排序、聚合、关联的文档数和拼写纠错等，本文介绍一个基于Elasticsearch实现的搜索建议。

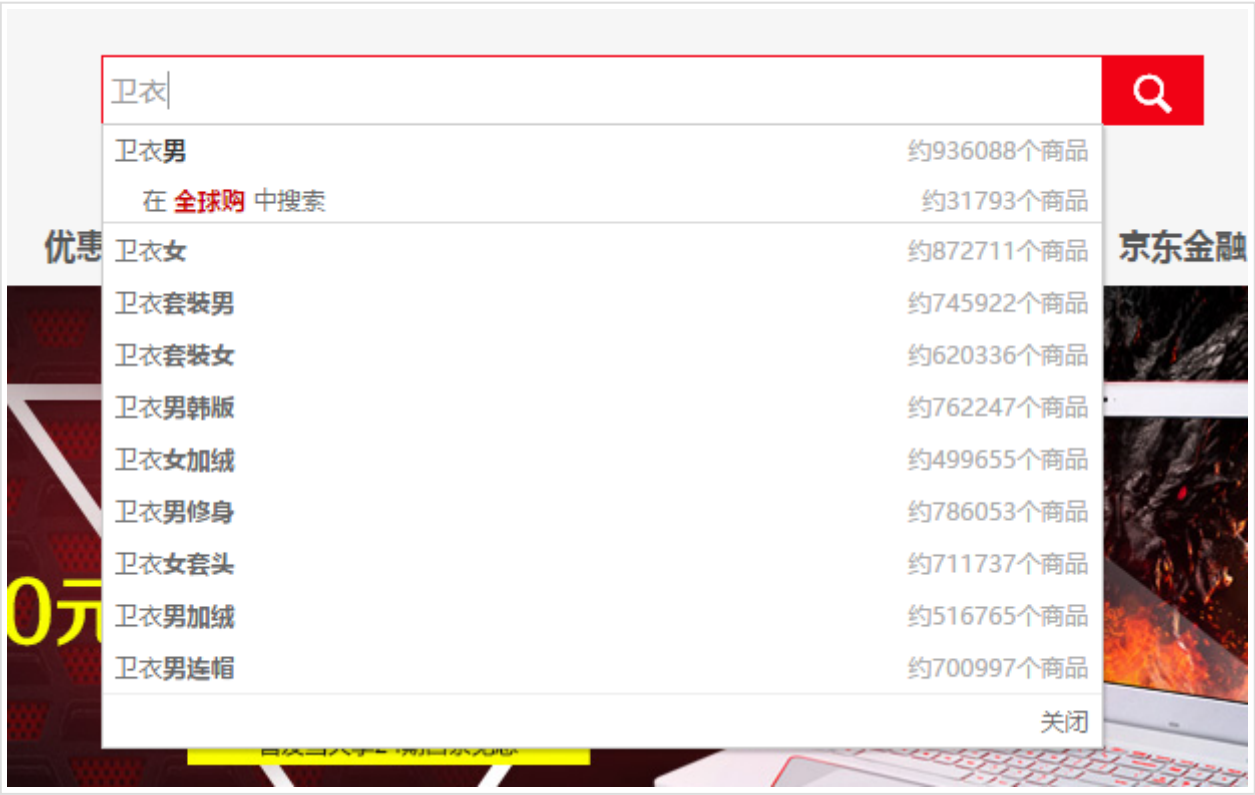
问题描述

电商网站的搜索是最基础最重要的功能之一，搜索框上面的良好体验能为电商带来更高的收益，我们先来看看淘宝、京东、亚马逊网站的搜索建议。

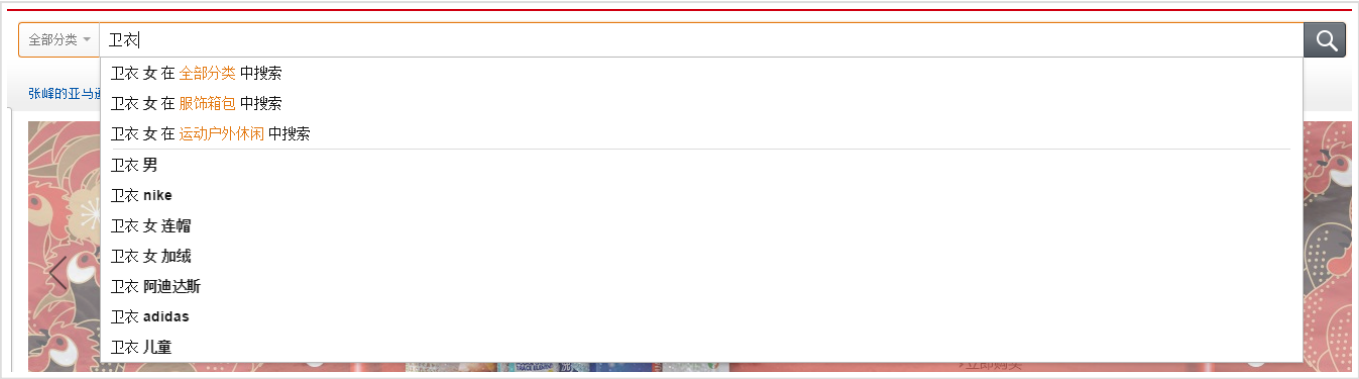
在淘宝的搜索框输入【卫衣】时，下方的搜索建议包括建议词以及相关的标签：



在京东的搜索框输入【卫衣】时，下方搜索建议右方显示建议词关联的商品数量：



在亚马逊的搜索框输入【卫衣】时，搜索建议上部分能支持在特定的分类下进行搜索：



通过上述对比可以看出，不同的电商对于搜索建议的侧重点略有不同，但核心的问题包括：

- 匹配：能够通过用户的输入进行前缀匹配；
- 排序：根据建议词的优先级进行排序；
- 聚合：能够根据建议词关联的商品进行聚合，比如聚合分类、聚合标签等；
- 纠错：能够对用户的输入进行拼写纠错；

搜索建议实现

在我们的搜索建议实现里，主要考虑了建议词的来源、匹配、排序、关联的商品数量和拼写纠错。

SuggestionDiscovery

- SuggestionDiscovery的职责是发现建议词；

- 建议词的来源可以是商品的分类名称、品牌名称、商品标签、商品名称的高频词、热搜词，也可以是一些组合词，比如“分类 + 性别”和“分类 + 标签”，还可以是一些自定义添加的词；
- 建议词维护的时候需要考虑去重，比如“卫衣男”和“卫衣 男”应该是相同的，“Nike”和“nike”也应该是相同的；
- 由于建议词的来源通常比较稳定，所以执行的周期可以比较长一点，比如每周一次；

SuggestionCounter

- SuggestionCounter的职责是获取建议词关联的商品数量，如果需要可以进行一些聚合操作，比如聚合分类和标签；
- SuggestionCounter的实现的时候由于要真正地调用搜索接口，应该尽量避免对用户搜索的影响，比如在凌晨执行并且使用单线程调用；
- 为了提升效率，应该使用Elasticsearch的Multi Search接口批量进行count，同时批量更新数据库里建议词的count值；
- 由于SuggestionCounter是比较耗资源的，可以考虑延长执行的周期，但是这可能会带来count值与实际搜索时误差较大的问题，这个需要根据实际情况考虑；

SuggestionIndexRebuilder

- SuggestionIndexRebuilder的职责是负责重建索引；
- 考虑到用户的搜索习惯，可以使用Multi-fields来给建议词增加多个分析器。比如对于【卫衣 套头】的建议词使用Multi-fields增加不分词字段、拼音分词字段、拼音首字母分词字段、IK分词字段，这样输入【weiyi】和【套头】都可以匹配到该建议词；
- 重建索引时是通过通过bulk批量添加到临时索引中，然后通过别名来更新；
- 重建索引的数据依赖于SuggestionCounter，因此其执行的周期应该与SuggestionCounter保持一致；

SuggestionService

- SuggestionService是真正处于用户搜索建议的服务类；
- 通常的实现是先到缓存中查询是否能匹配到缓存记录，如果能匹配到则直接返回；否则的话调用Elasticsearch的Prefix Query进行搜索，由于我们在重建索引的时候定义了Multi-fields，在搜索的时候应该用boolQuery来处理；如果此时Elasticsearch返回不为空的结果数据，那么加入缓存并返回即可；

```
POST /suggestion/_search
{
  "from" : 0,
  "size" : 10,
  "query" : {
    "bool" : {
      "must" : {
        "bool" : {
```

```

    "should" : [ {
      "prefix" : {
        "keyword" : "卫衣"
      }
    }, {
      "prefix" : {
        "keyword.keyword_ik" : "卫衣"
      }
    }, {
      "prefix" : {
        "keyword.keyword_pinyin" : "卫衣"
      }
    }, {
      "prefix" : {
        "keyword.keyword_first_py" : "卫衣"
      }
    } ]
  }
},
"filter" : {
  "range" : {
    "count" : {
      "from" : 5,
      "to" : null,
      "include_lower" : true,
      "include_upper" : true
    }
  }
}
},
"sort" : [ {
  "weight" : {
    "order" : "desc"
  }
}, {
  "count" : {
    "order" : "desc"
  }
} ]
}
}

```

- 如果Elasticsearch返回的是空结果，此时应该需要增加拼写纠错的处理（拼写纠错也可以在调用Elasticsearch搜索的时候带上，但是通常情况下用户并没有拼写错误，所以建议还是在后面单独调用suggester）；如果返回的suggest不为空，则根据新的词调用建议词服务；比如用户输入了【adidss】，调用Elasticsearch的suggester获取到的结果是【adidas】，则再根据adidas进行搜索建议词处理。

```

POST /suggestion/_search
{

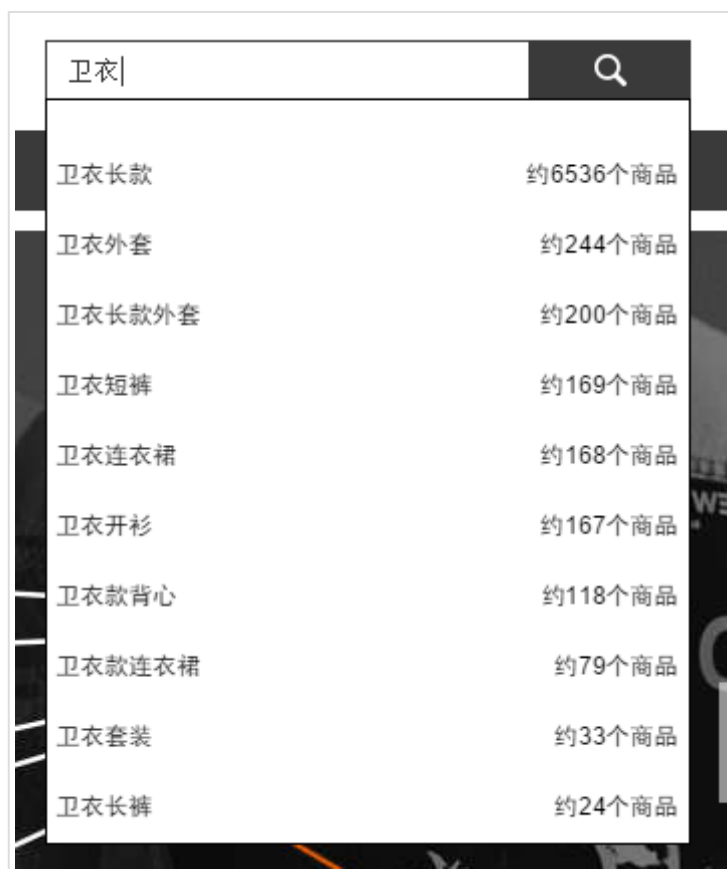
```

```
"size" : 0,
"suggest" : {
  "keyword_suggestion" : {
    "text" : "adidss",
    "term" : {
      "field" : "keyword",
      "size" : 1
    }
  }
}
```

- 关于排序：在我们的实现里面是通过weight和count进行排序的，weight目前只考虑了建议词的类型（比如分类 > 品牌 > 标签）；

实现效果和后续改进

- 通过上面的实现，我们已经能实现一个比较强大的搜索建议词了，实际的效果如下所示：



- 后续可以考虑的改进：参考亚马逊增加分类的聚合展示、增加用户个性化的处理支持更好的建议词排序、基于用户的搜索历史支持更好的建议词推荐；

参考资料

- [Elasticsearch Prefix Query](#)

- [Elasticsearch Suggester](#)



扫一扫，关注我的微信公众号

Elasticsearch # 搜索建议

◀ 《深入理解Java虚拟机》读书笔记5：类加载机制与字节码执行引擎

《深入理解Java虚拟机》读书笔记6：程序编译与代码优化 ▶

4

© 2016 - 2017 ♥ Gino Zhang

由 [Hexo](#) 强力驱动 | 主题 - [NexT.Mist](#)

👤 访问人数 12461 | 👁 总访问量 24704