随着央视诗词大会的热播，小史开始对诗词感兴趣，最喜欢的就是飞花令的环节。

但是由于小史很久没有背过诗词了，飞一个字很难说出一句，很多之前很熟悉的诗句也想不起来。


请说出带“前”字的诗句。
微微一笑
小史
吕老师


啊？想不出来。
努力思考
小史
吕老师


那小史，你背一下李白的静夜思。
微微一笑
小史
吕老师


那简单啊，床前明月光，疑是地上霜……
等等，这句就有前啊！哎呀！
懊悔不已
小史
吕老师


小史，你再背一下李白的望庐山瀑布。
微微一笑
小史
吕老师


那简单啊，日照香炉生紫烟，遥看瀑布挂前川……等等，这句也有前啊！哎呀呀！
懊悔不已
小史
吕老师


哈哈，其实你都会的，只是想不起来而已。
微微一笑
小史
吕老师


哎，为什么会这样呢?
疑惑不解
小史
吕老师

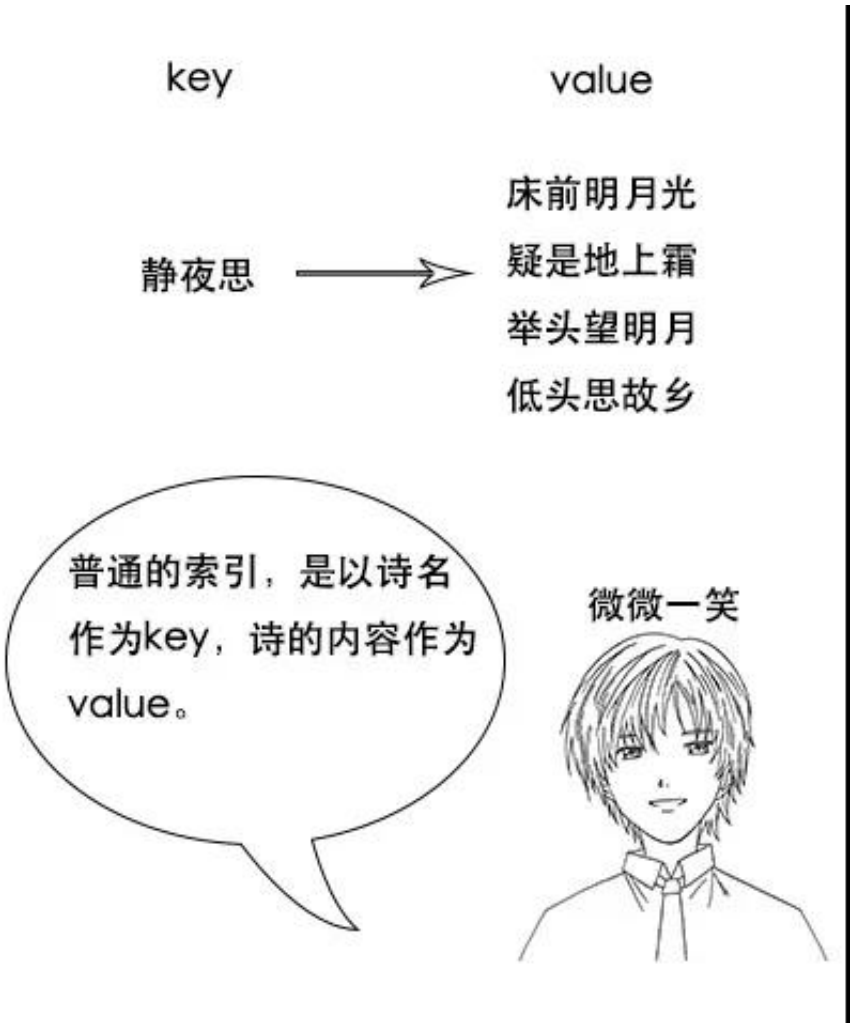

因为你的脑子里并没有建立倒排索引。
微微一笑
小史
吕老师


蛤？啥是倒排索引？
提出疑问
小史
吕老师


小史，你平时背诗的时候是怎么背的呢？
微微一笑
小史
吕老师


从前往后啊，先记诗名、作者，然后背诗的内容。
不假思索
小史
吕老师


没错，所以在你的脑子里，索引是这样的。
微微一笑
小史
吕老师


key
value
静夜思
床前明月光
疑是地上霜
举头望明月
低头思故乡
普通的索引，是以诗名作为key，诗的内容作为value。
微微一笑


所以我让你背静夜思你马上能反应过来，因为你从索引直接找到了诗。
微微一笑
小史
吕老师

吕老师：但是我让你说出带“前”字的诗句，由于没有索引，你只能遍历脑海中所有诗词，当你的脑海中诗词量大的时候，就很难在短时间内得到结果了。


你这么一说好像确实是这样，那倒排索引又是啥?
提出疑问
小史
吕老师


倒排索引又叫反向索引，我举个例子你就明白了，比如这样建立索引。
微微一笑
小史
吕老师


key
value
前
床前明月光
疑是地上霜
倒排索引，以“前”字作为key，诗句内容作为value。
微微一笑

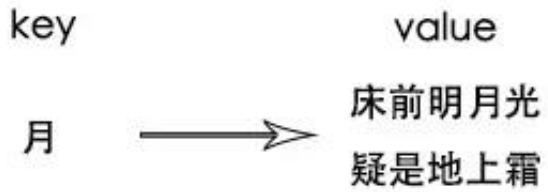
你的意思是“前”作为索引?
瞬间理解
小史
吕老师


没错，这样的话，让你背带“前”字的诗，你是不是马上就能背出来了?
微微一笑
小史
吕老师


哦，明白了，这样记的话，确实可以很快找到带“前”字的诗句。
瞬间理解
小史
吕老师

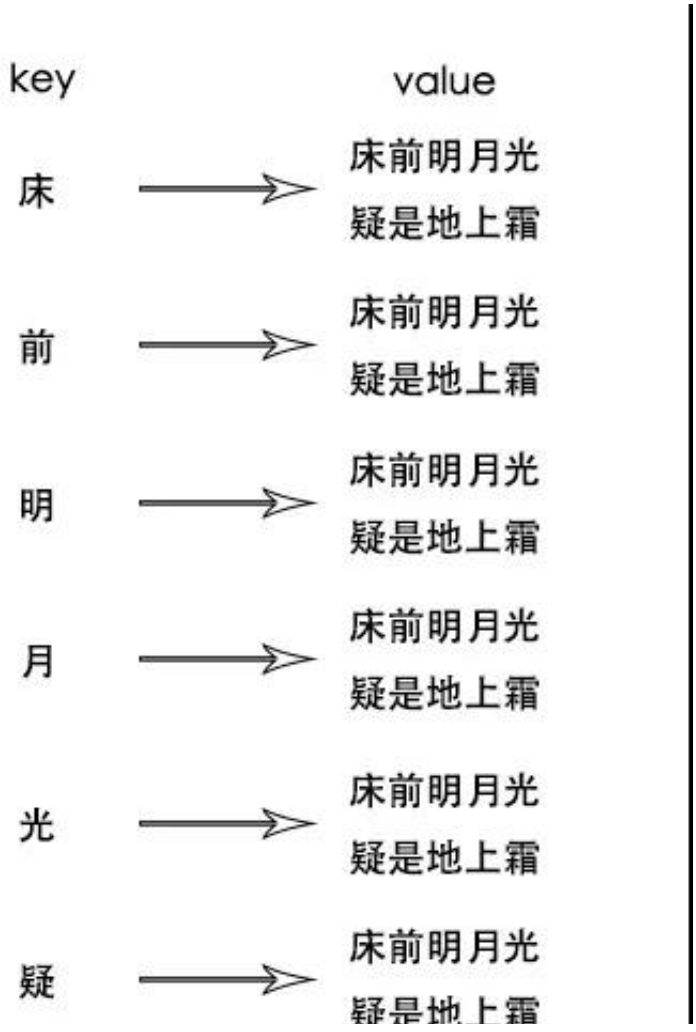
索引量爆炸


不过吕老师，我想到一个问题啊，“床前明月光 疑是地上霜”这句，也可以以“月”字建立索引。
提出疑问
小史
吕老师


key
value
月
床前明月光
疑是地上霜


这句诗以“月”字建立索引
也完全没有违和感啊


确实是，实际上，这一句就可以建立10个索引。
微微一笑
小史
吕老师


key
value
床
床前明月光
疑是地上霜
前
床前明月光
疑是地上霜
明
床前明月光
疑是地上霜
月
床前明月光
疑是地上霜
光
床前明月光
疑是地上霜
疑
床前明月光

是 → 床前明月光
疑是地上霜

地 → 床前明月光
疑是地上霜

上 → 床前明月光
疑是地上霜

霜 → 床前明月光
疑是地上霜


对呀，本来正向只有1个索引，反向却有10个索引，这样下去记忆量会爆炸性增长吧?
提出疑问
小史
吕老师


没错，反向索引的建立，数据量确实会更多，如果你的文章越长，索引可能会越多。
微微一笑
小史
吕老师


那还能不能愉快的背诗了，这样下去脑细胞都不够了。
心生抱怨
小史
吕老师


只要开动脑筋，凡事都是有办法的，小史，难道数据就不能压缩一下吗?
微微一笑
小史
吕老师

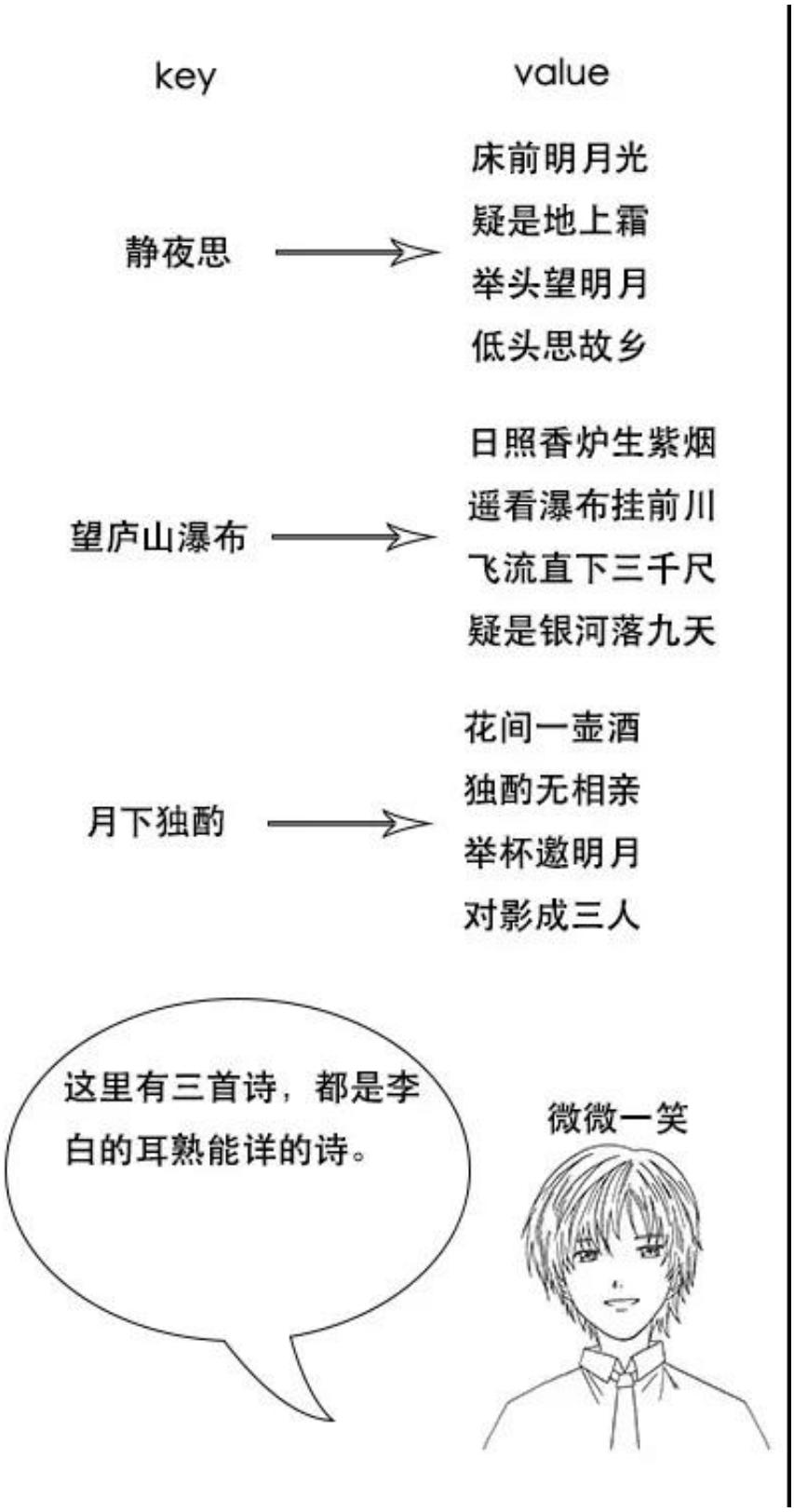

怎么压缩?
提出疑问
小史
吕老师


既然你已经可以通过诗名就想起一首诗，那反向索引就没必要索引到诗句了，只要索引到诗名就行。
微微一笑
小史
吕老师

key		value
床	→	静夜思
前	→	静夜思
明	→	静夜思
月	→	静夜思
光	→	静夜思
疑	→	静夜思
是	→	静夜思
地	→	静夜思
上	→	静夜思
霜	→	静夜思


哦，我明白了，这样的话，数据量确实减少一些了。
瞬间理解
小史
吕老师


没错，但是这还只是一首诗的情况，多首诗咱们还可以形成索引矩阵。比如这样几首诗
微微一笑
小史
吕老师


key
value
静夜思
床前明月光
疑是地上霜
举头望明月
低头思故乡
望庐山瀑布
日照香炉生紫烟
遥看瀑布挂前川
飞流直下三千尺
疑是银河落九天
月下独酌
花间一壶酒
独酌无相亲
举杯邀明月
对影成三人
这里有三首诗，都是李白的耳熟能详的诗。
微微一笑

其中，静夜思、望庐山瀑布都有“前”字，而静夜思、月下独酌都有“月”字，他们之间的关系大概是这样的

微微一笑

小史

吕老师

静夜思 ——————— 带“前”字

望庐山瀑布 ——————— 带“前”字

静夜思 ——————— 带“月”字

月下独酌 ——————— 带“月”字

三首诗与“前”字和“月”字的关系分别如上图所示。

微微一笑


所以建立索引的时候可以这样
微微一笑
小史
吕老师


key
value
前
静夜思
望庐山瀑布
月
静夜思
月下独酌
“前”字可以索引到两首诗，“月”字也可以索引到两首诗。
微微一笑

搜索引擎原理


小史，有没有感觉你这个过程和咱们百度或者谷歌搜索类似呀?
微微一笑
小史
吕老师


哦，确实是哦，都是根据一个内容，搜索到要找的文章。
努力思考
小史
吕老师

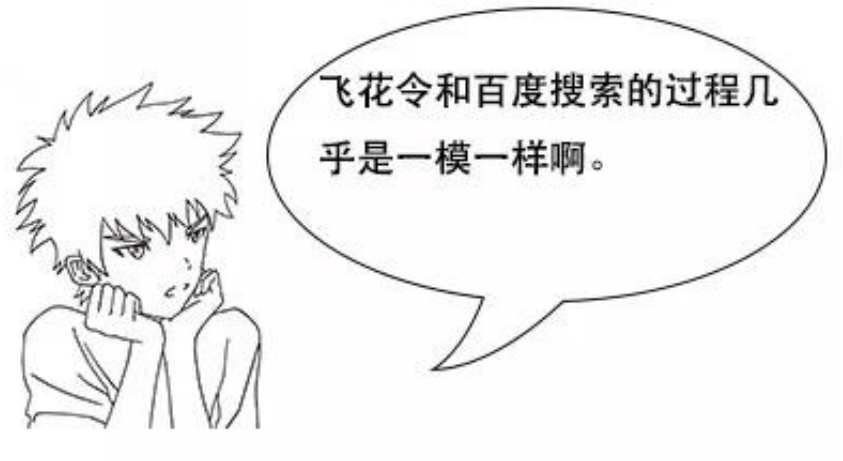
飞花令

说出带“前”的诗句

床前明月光

遥看瀑布挂前川

百度搜索




明白了，原来如此，所以它们才能快速命中要搜索的内容。
瞬间理解
小史
吕老师


对，只不过它们的流程稍微复杂一点，包括网页爬取，停顿词过滤等等。
微微一笑
小史
吕老师


网页爬取我知道啊，就是所谓的爬虫嘛，
停顿词过滤是什么意思?
提出疑问
小史
吕老师


比如你爬到一篇文章，里面有一段话
微微一笑
小史
吕老师

面试官的每一个问题，都是有考察点的。但是对于这样的软问题，你可以好好回答来主导方向，要体现你的深度思考，体现你在工作中如何创造价值，而不是浮于问题表面。

的、而这种没有意义的词可以认为是停顿词


哦哦，就好像就是所谓的“分词”？
瞬间理解
小史
吕老师


哟，小史，看来你对这方面有过研究啊。
微微一笑
小史
吕老师


嗯，之前看别的文章的时候有看到过。
所以搜索引擎都是对文章分词之后，再根据关键字建立倒排索引咯。
细心总结
小史
吕老师

1、爬取

面试软技巧

面试官的每一个问题，都是有考察点的。但是对于这样的软问题，你可以好好回答来主导方向，要体现你的深度思考，体现你在工作中如何创造价值，而不是浮于问题表面。

2、分词

面试官
软问题
考察
回答
深度思考
创造价值
……

3、建立反向索引

面试官 ⟶ 面试软技巧
软问题 ⟶ 面试软技巧
考察 ⟶ 面试软技巧
回答 ⟶ 面试软技巧
深度思考 ⟶ 面试软技巧
创造价值 ⟶ 面试软技巧

Elasticsearch 简介


吕老师，如果我要自己实现一个搜索引擎，该怎么做呢？
提出疑问
小史
吕老师


一个搜索引擎的实现工作量是巨大的，但是你其实没有必要自己实现，业界已经有成熟的开源解决方案了。
微微一笑
小史
吕老师


果然已经有“轮子”了，给我讲讲吧。
虚心求教
小史
吕老师


其实很早以前，业界有一个叫做lucene的库，用它就可以方便地建立倒排索引。
微微一笑
小史
吕老师

吕老师：但是 Lucene 还是一个库，必须要懂一点搜索引擎原理的人才能用的好，所以后来又有人基于 Lucene 进行封装，写出了 Elasticsearch。


elasticsearch又有哪些好处呢?
提出疑问
小史
吕老师


elasticsearch将对搜索引擎的操作都封装成了restful的api，通过http请求就能对其进行操作。
微微一笑
小史
吕老师


同时，它还考虑了海量数据，实现了分布式，是一个可以存储海量数据的分布式搜索引擎。
微微一笑
小史
吕老师


那它也是基于hdfs的吗?
提出疑问
小史
吕老师


这倒不是，elasticsearch和hadoop系列不是一路人，它是完完全全自己的一套。
微微一笑
小史
吕老师


哦，原来如此，吕老师，你给我简单介绍介绍吧。
虚心求教
小史
吕老师

Elasticsearch 基本概念

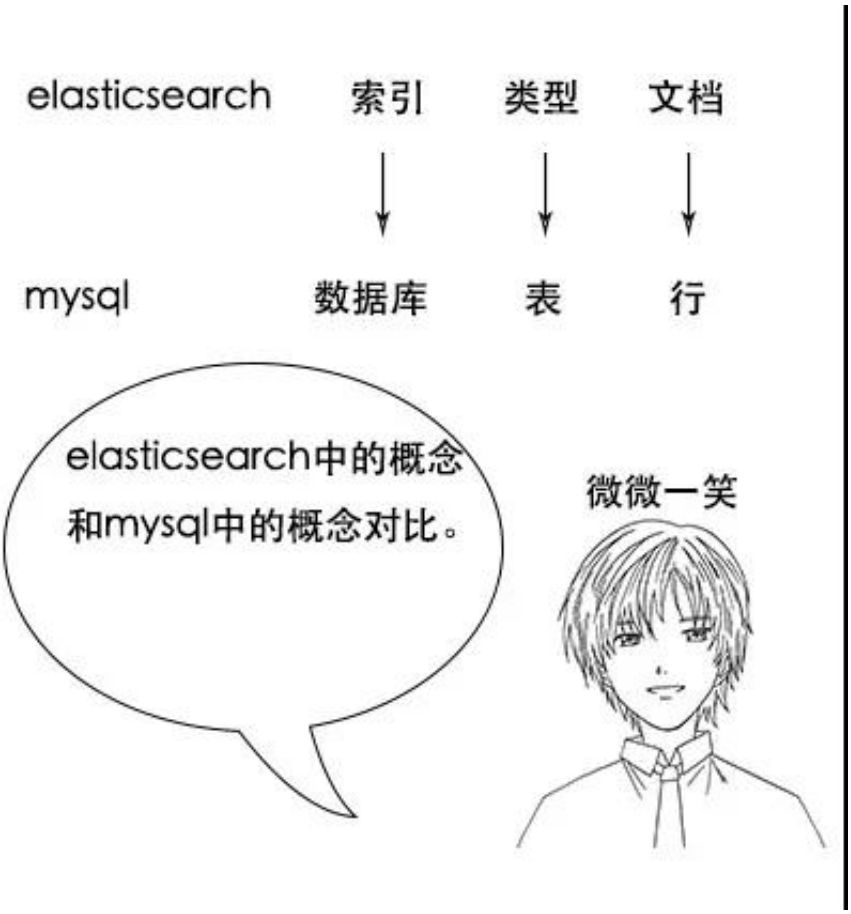

索引？不就是刚刚说的key吗？
不假思索
小史
吕老师


此索引非彼索引，elasticsearch中的索引是存放数据的地方，你可以理解为mysql中的一个数据库。
微微一笑
小史
吕老师


哦，我明白了，相当于是elasticsearch中的一个概念。那类型和文档又是什么呢？
瞬间理解
小史
吕老师


类型是用来定义数据结构的，你可以认为是mysql中的一张表。文档就是最终的数据了，你可以认为一个文档就……
微微一笑
小史
吕老师

吕老师：类型是用来定义数据结构的，你可以认为是 MySQL 中的一张表。文档就是最终的数据了，你可以认为一个文档就是一条记录。



吕老师：比如一首诗，有诗题、作者、朝代、字数、诗内容等字段，那么首先，我们可以建立一个名叫 Poems 的索引，然后创建一个名叫 Poem 的类型，类型是通过 Mapping 来定义每个字段的类型。

比如诗题、作者、朝代都是 Keyword 类型，诗内容是 Text 类型，而字数是 Integer 类型，***就是把数据组织成 Json 格式存放进去了。

```
索引
poems

类型
"poem": {
  "properties": {
    "title": {
      "type": "keyword",
    },
    "author": {
      "type": "keyword",
    }
```

```
        },
        "dynasty": {
            "type": "keyword"
        },
        "words": {
            "type": "integer"
        },
        "content": {
            "type": "text"
        }
    }
}
```

文档

```
{
    "title": "静夜思",
    "author": "李白",
    "dynasty": "唐",
    "words": "20",
    "content": "床前明月光，疑是地上霜。举头望明月，低头思故乡。"
}
```


keyword和text都是表示字符串吧？它们有啥区别呢？
提出疑问
小史
吕老师


这个问题问得好，这涉及到分词的问题，keyword类型是不会分词的，直接根据字符串内容建立反向索引，text类型……
微微一笑
小史
吕老师

吕老师：这个问题问得好，这涉及到分词的问题，Keyword 类型是不会分词的，直接根据字符串内容建立反向索引，Text 类型在存入 Elasticsearch 的时候，会先分词，然后根据分词后的内容建立反向索引。

吕老师：之前我们说过，Elasticsearch 把操作都封装成了 HTTP 的 API，我们只要给 Elasticsearch 发送 HTTP 请求就行。

比如使用 curl -XPUT 'http://ip:port/poems'，就能建立一个名为 Poems 的索引，其他操作也是类似的。