# 搜索引擎基本原理

小龙

# 用户输入”六味丸“，如何找到“六味地黄丸”这个产品？

# 数据库like

- **select * from sku where name like ‘%六%’ and name like ‘%味%’ and name like ‘%丸%’**
- 存在什么问题?

# 倒排索引

| | | | |
|---|---|---|---|
| 六 | 1 | 2 | 5 |
| 味 | 1 | 5 | 8 |
| 地 | 2 | 5 | 6 |
| 黄 | 1 | 5 | 7 |
| 丸 | 1 | 5 | 9 |

- 六和塔下，小丸子在津津有味的吃东西
- 吃了六种肉丸子，味道还不错
- 九芝堂六味地黄丸很给力
- 搜索“六味丸”，如何只搜到最后一条？

# 分词

| | | |
|---|---|---|
| 六和塔 | 1 | |
| 六味 | 3 | |
| 地黄 | 3 | |
| 肉丸 | 2 | |
| 丸 | 3 | |
| 小丸子 | 1 | |
| 六 | 1 | |

- 今年我要去魔都跑马拉松
- 魔都今天雾霾好严重
- 北京真是个鬼地方
- 搜索“上海”时，如何能搜到前面两句？

# 同义词

| | | |
|---|---|---|
| 魔都 | 1 | 2 |
| 上海 | 1 | 2 |
| 北京 | 3 | |
| | | |

- 春天来了，小草发芽了
- 你喜欢春天还是夏天？我喜欢夏天。
- 风和日丽的日子里，蜜蜂出来采蜜了
- 搜索“春天 蜜蜂”时，如何排序？

# TF-IDF

- TF 词频，出现的频率越大，词越重要
- IDF 逆文档频率，词越少见，词越重要
- TF * IDF 说明词在文档里的重要性

# 向量空间模型

- d1 = (1, 0)
- d2 = (1, 0)
- d3 = (0, 1.2)
- q = (1, 1)
- max(q * d1, q * d2, q * d3) = 1.2 所以d3排最前面
- 相关性排序的水很深。。。

# 开源搜索引擎

- Lucene 索引库(Java)
- Solr(基于Lucene)
- Elasticsearch(基于Lucene)(推荐使用)
- Sphinx(C++)

# 参考链接

- Lucene: https://lucene.apache.org/
- Solr: http://lucene.apache.org/solr/
- Elasticsearch: https://www.elastic.co/
- Sphinx: http://sphinxsearch.com/

# 谢谢观赏 Q&A