III Inverted File Index

> 内容比较零碎，感觉量化内容很少，都是思路介绍

Introduction

How can I find in which retrieved web pages that include "Computer Science"?

【Definition】 Index is a mechanism for locating a given term in a text.

【Definition】 Inverted file contains a list of pointers (e.g. the number of a page) to all occurrences of that term in the text.

Doc	Text
1	Gold silver truck
2	Shipment of gold damaged in a fire
3	Delivery of silver arrived in a silver truck
4	Shipment of gold arrived in a truck

Inverted File Index

No.	Term	Times; Documents
1	a	<3; 2,3,4>
2	arrived	<2; 3,4>
3	damaged	<1; 2>
4	delivery	<1; 3>
5	fire	<1; 2>
6	gold	<3; 1,2,4>
7	of	<3; 2,3,4>
8	in	<3; 2,3,4>
9	shipment	<2; 2,4>
10	silver	<2; 1,3>
11	truck	<3; 1,3,4>

Build The Table

```
while ( read a document D ) {
    while ( read a term T in D ) {
        if ( Find( Dictionary, T ) == false )
            Insert( Dictionary, T );
        Get T's posting list;
        Insert a node to T's posting list;
    }
}
Write the inverted index to disk;
```

这只是最粗糙的一种形式

那么 **Term** 怎么定义呢?

用到两个技术

- Word Stemming (词源分析)

  Process a word so that only its stem or root form is left

- Stop Words (停用词)

  Some words are so common that almost every document contains them, such as “a” “the” “it”. It is useless to index them. They are called stop words. We can eliminate them from the original documents.

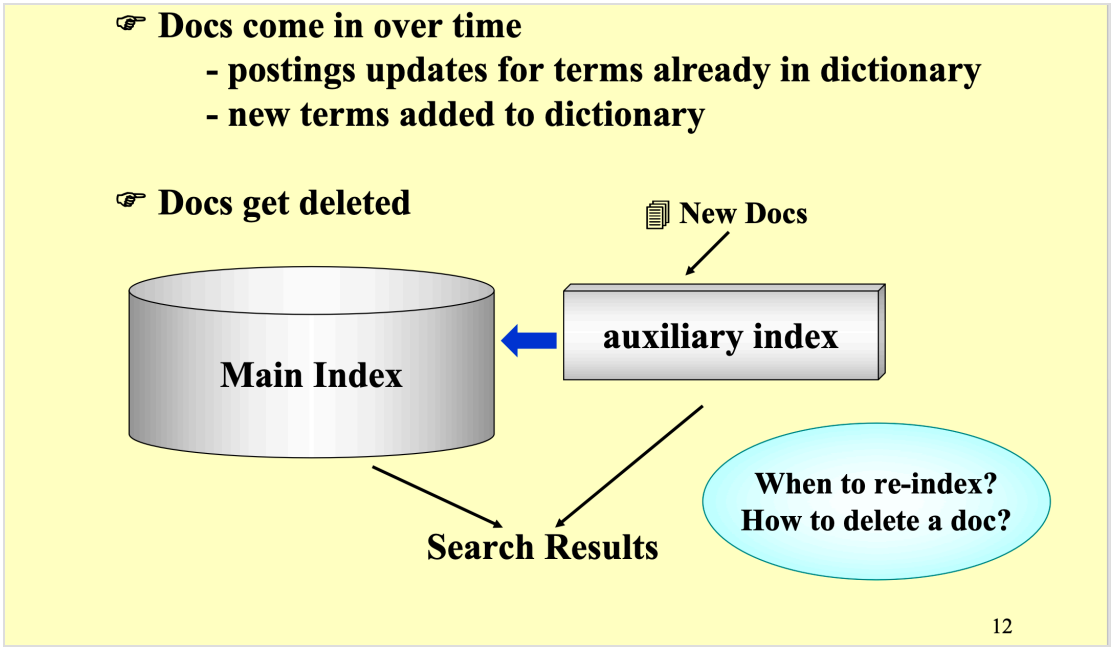
Distributed indexing

有两种分布式的策略，其一是根据单词的字典序进行分布式，其二是根据文档进行分布式

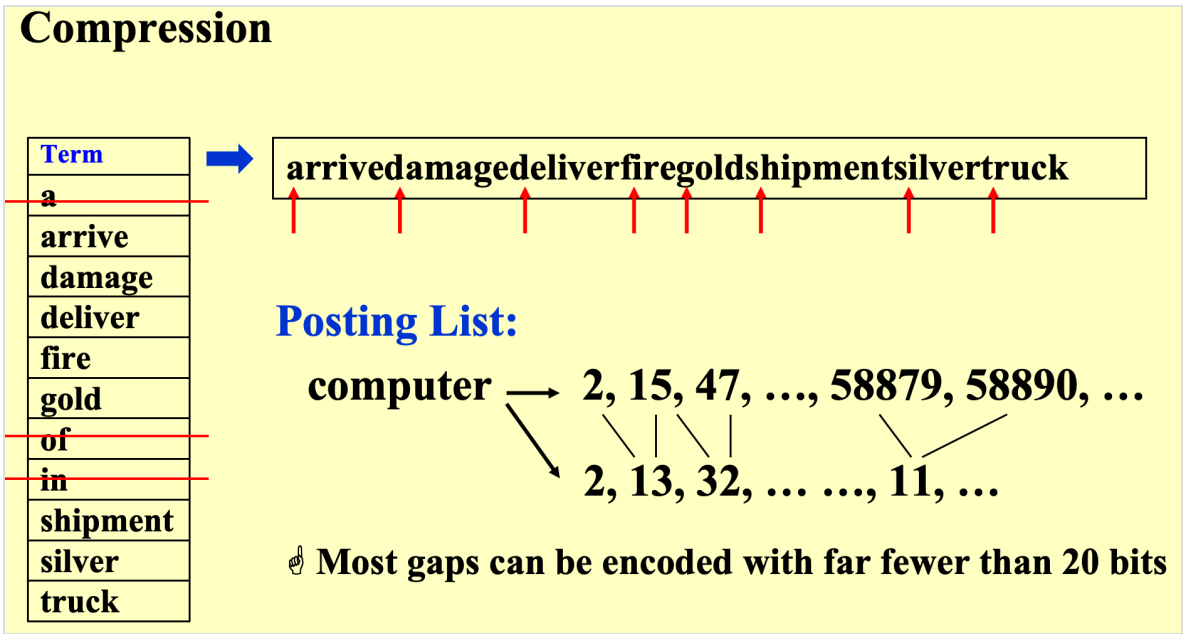
> 显然根据单词的内容进行分布式，能够提高索引效率，但是这样的话，我们就需要将所有形式接近的单词都存储在一个地方，这样就会造成单点故障，容灾能力很差，所以这种方式并不是很好。而第二种办法则有较强的容灾性能。即使一台机器无法工作，也不会剧烈影响到整个系统的工作

Dynamic indexing

> 没看懂想强调什么特殊的



Index Compression And Index Thresholding

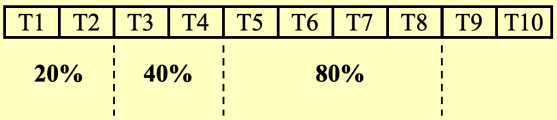


Thresholding

☞ **Document: only retrieve the top *x* documents where the documents are ranked by weight**

- 👎 **Not feasible for Boolean queries**
- 👎 **Can miss some relevant documents due to truncation**

☞ **Query: Sort the query terms by their frequency in ascending order; search according to only some percentage of the original query terms**



14

Measures for a search engine

> 人工智能入门必备小知识😋

- Data Retrieval Performance Evaluation (after establishing correctness)
  - Response time
  - Index space
- Information Retrieval Performance Evaluation
  - How **relevant** is the answer set?

混淆矩阵

图中的表格展示了经典的**信息检索混淆矩阵**:

	相关文档	不相关文档	总计
被检索到	R_R	I_R	检索到的文档总数
未被检索到	R_N	I_N	未检索到的文档总数

其中:

- R_R：被正确检索到的相关文档数（True Positives）
- I_R：被错误检索到的不相关文档数（False Positives）
- R_N：未被检索到的相关文档数（False Negatives）
- I_N：未被检索到的不相关文档数（True Negatives）

$$P = \frac{R_R}{R_R + I_R}$$

解释：在所有被检索到的文档中，真正相关的文档所占比例

- 关注**检索结果的质量**
- 值越高说明"垃圾结果"越少

$$R = \frac{R_R}{R_R + R_N}$$

解释：在所有相关文档中，被成功检索到的文档所占比例

- 关注**检索系统的覆盖度**
- 值越高说明"漏检"越少
- 高精确率：严格筛选，只返回最有把握的结果 → 可能漏掉一些相关文档
- 高召回率：宽松检索，尽量不遗漏相关文档 → 可能混入更多不相关文档

fafaf