

#ADS03 Inverted File Index

1. Term-Document Incidence Matrix

2. Inverted File Index

Index Generator

Distributed indexing

Dynamic indexing

Measures for a search engine

3. Exercise

To find web pages on Internet.

1. Term–Document Incidence Matrix

[(Example)] Document sets

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

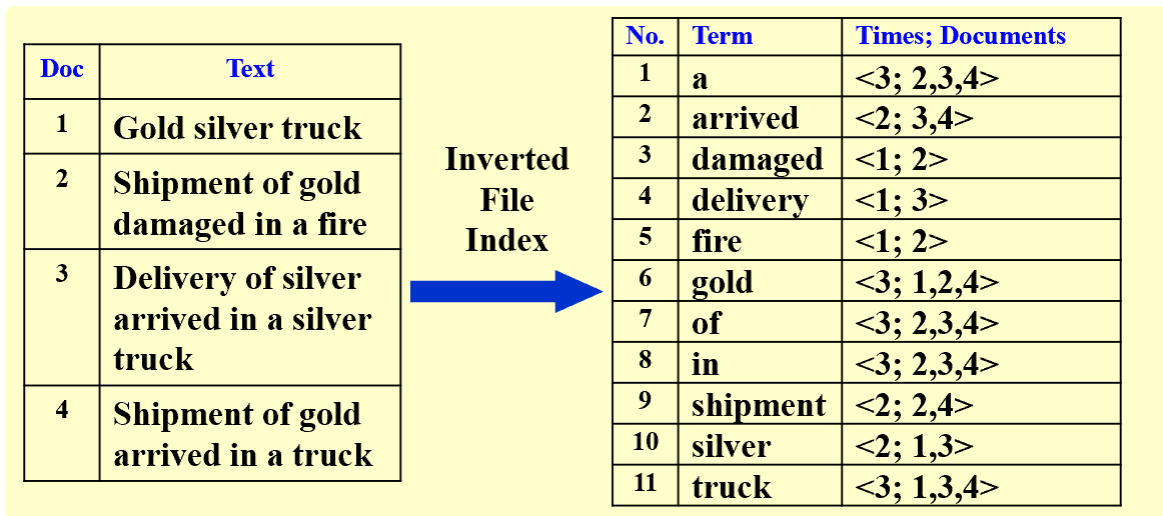
	1	2	3	4
a	0	1	1	1
arrived	0	0	1	1
damaged	0	1	0	0
delivery	0	0	1	0
fire	0	1	0	0
gold	1	1	0	1
of	0	1	1	1
in	0	1	1	1
shipment	0	1	0	1
silver	1	0	1	0
truck	1	0	1	1

silver & truck

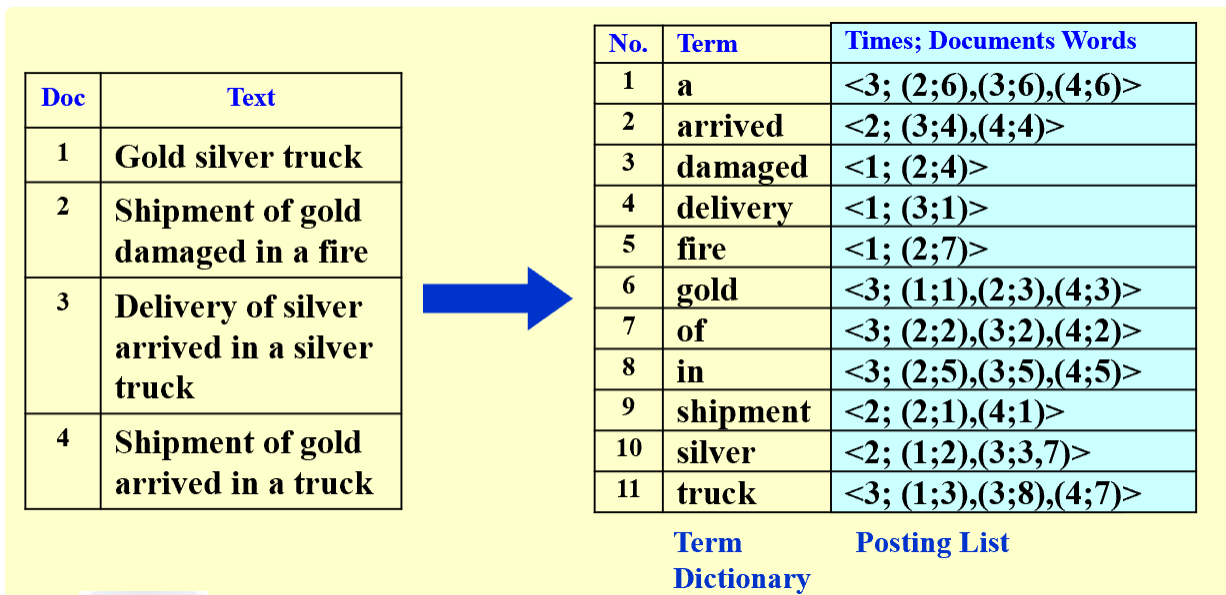
2. Inverted File Index

[(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.



To easily print the sentences which contain the words and highlight the words:



Index Generator

```
1 while(read a document D){
2     while(read a term T in D){ //1.Token Analyzer Stop Filter
3         if(Find (Dictionary, T) == False) // 2.Vocabulary Scanner
4             Insert ( Dictionary, T); //3.Vocabulary Insertor
5         Get T's posting list;
6         Insert a node to T's posting list;
7     }
8 }
9 write the inverted index to disk; // 4.Memory management
```

posting list : 倒排列表，记录了出现过某个单词的所有文档的文档列表以及单词在该文档中出现的位置。

1. Token Analyzer Stop Filter
2. Vocabulary Scanner
3. Vocabulary Insertor
4. Memory management

- 从文件中读取词
- 将该词提取为词干(word stemming)，即去除第三人称形式、过去式、进行时等形式，留下词干），并去除分词(stop word)，即“a”，“is”等没有意义的词
- 检查该词是否已经在词典之中。
- 若不在，则将该词添加入词典之中。更新索引信息。
- 建立完毕后，将索引文件存入磁盘。

When reading a term

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

- Stop words

Some words are too common. It is useless to index them.
Eliminate.

When accessing a term

- Search trees
- Hashing

While not having enough memory

▼ Insert Index

C | 复制代码

```
1  BlockCnt = 0;
2  while(read a document D){
3      while(read a term T in D){ //1
4          if(out of memory){
5              Write BlockInde[BlockCnt] to disk;
6              BlockCnt++;
7              FreeMemory;
8          }
9          if(Find (Dictionary, T) == False) // 2
10             Insert ( Dictionary, T); //3
11             Get T's posting list;
12             Insert a node to T's posting list;
13         }
14     }
15     for(i=0;i<BlockCnt;i++)
16         Merge(InvertedIndex, BlockIndex[i]);
```

Distributed indexing

当倒排索引文件较大时，涉及到倒排索引分布式存储技术。

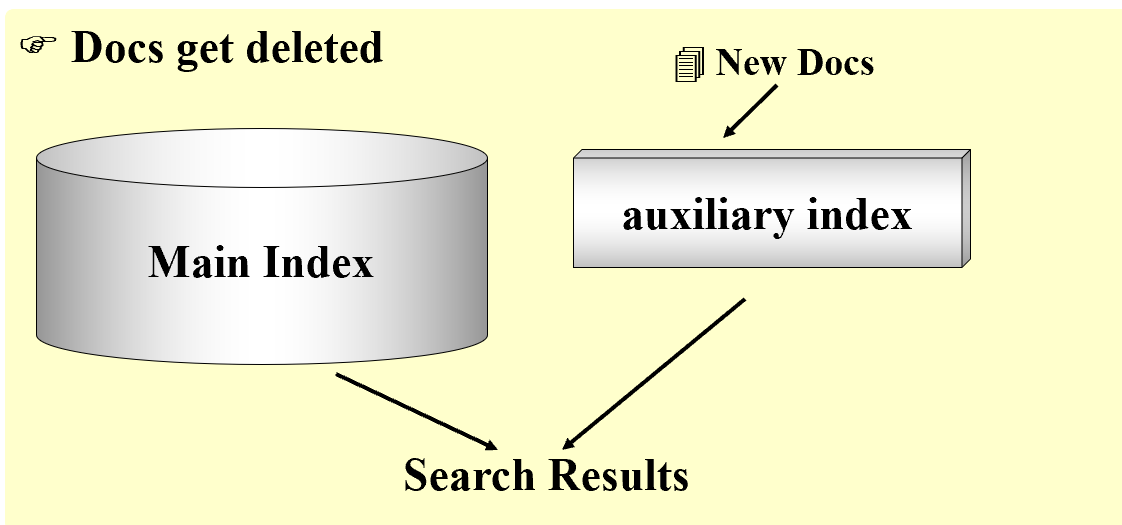
前者查找性能更高，但后者更可靠。

Term-partitioned index

Document-partitioned index

Dynamic indexing

- Docs come in over time
 - postings updates for terms already in dictionary
 - new terms added to dictionary
- Docs get deleted



由于需要被索引的文档集可能是动态变化的（例如添加新文档、删除现有文档），因此索引需要适应这种变化。

最简单的更新办法是周期性地对文档集从头开始进行索引重构。如果要求能够及时检索到新文档，那么一种方法是同时保持两个索引：一个是大的主索引，保存在磁盘中，另一个是小的用于存储新文档信息的辅助索引，辅助索引保存在内存中。检索时可以同时遍历两个索引并将结果合并。而文档的删除记录在一个无效位向量中，在返回检索结果之前可以利用它过滤掉已经删除的文档。每当辅助索引变得很大时，就将它合并到主索引中。

Compression

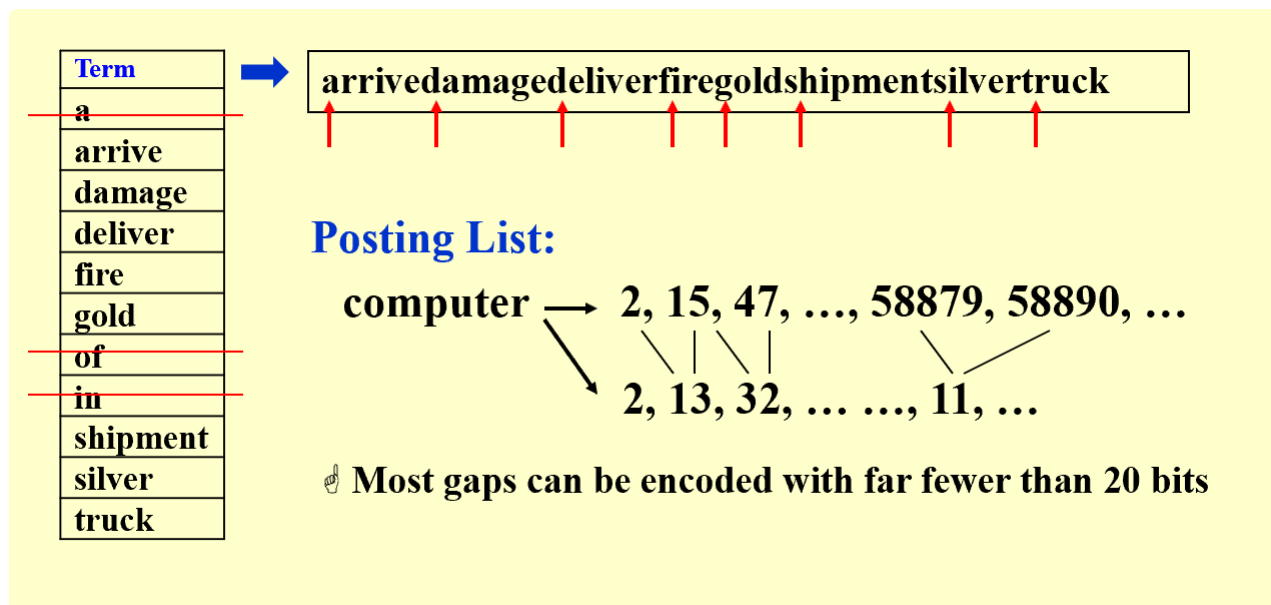
索引文件压缩

一般来说，对索引文件进行压缩不但可以减小空间，并且可以提高索引效率。这是因为，采用高效的压缩算法，虽然将耗费一定时间在内存中进行解压，但因为能提高cache的利用率，并能提高从磁盘到内存的读取效率，所以总体来说效率将得到提升。

索引文件压缩的内容在 高级数据结构课程 课件中提到了两种实现方式：

一是将词典看为单一字符串，以消除用定长方法来存储单词所存在的空间浪费；

二是docID的存储只记录与上一项docID的差值来减少docID存储长度。



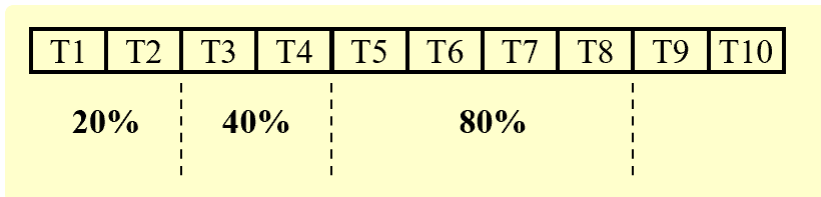
Thresholding

在现实使用中人们往往只关心结果中的前一部分，所以搜索时可以通过只搜索前x%来提高效率。

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



Measures for a search engine

- How fast does it index
 - Number of documents / hour
- How fast does it search
 - Latency as a function of index size
- Expressiveness of query language
 - Ability to express complex information needs
 - Speed on complex queries
- User happiness?
 - Data Retrieval Performance Evaluation (after establishing correctness)
 - response time
 - index space
 - Information Retrieval Performance Evaluation
 - How relevant is the answer set?

Relevant measurement requires 3 elements:

- A benchmark document collection
- A benchmark suite of queries
- A binary assessment of either Relevant or Irrelevant for each query–doc pair

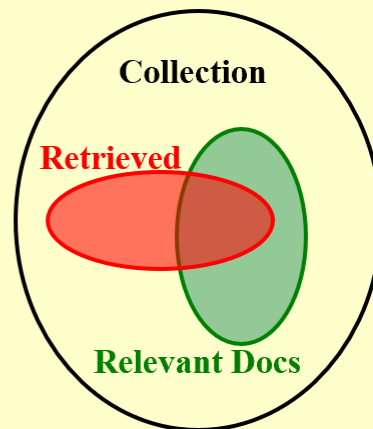
benchmark : 基准

除了响应时间、索引文件大小以外，主要从精确度Precision和召回度Recall进行衡量。

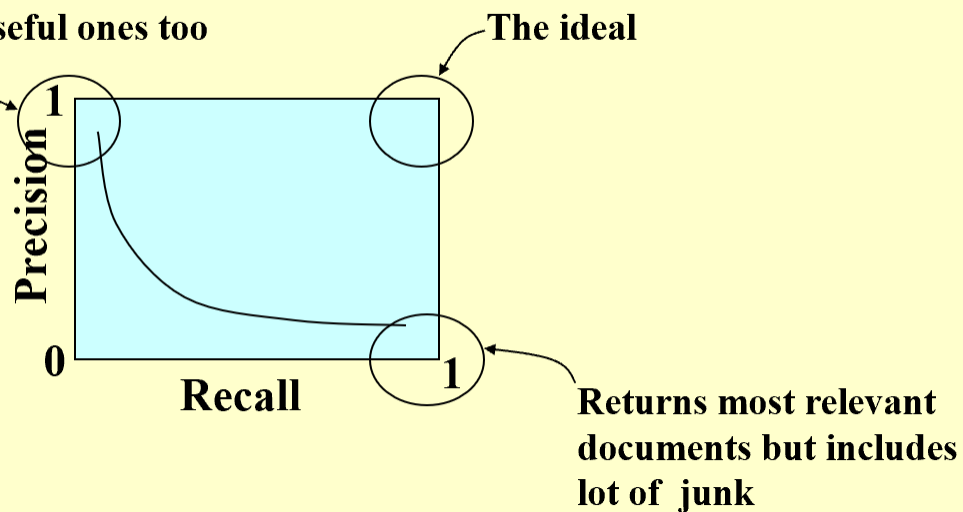
	Relevant	Irrelevant
Retrieved	R_R	I_R
Not Retrieved	R_N	I_N

Precision $P = R_R / (R_R + I_R)$

Recall $R = R_R / (R_R + R_N)$



Returns relevant documents but misses many useful ones too



3. Exercise

1


1-1 分数 1

作者 陈越 单位 浙江大学

In distributed indexing, document-partitioned strategy is to store on each node all the documents that contain the terms in a certain range.

☐ T ☒ F

答案正确: 1 分

 创建提问

是按文档编号进行的分类。

2

1-2 分数 1

作者 陈越 单位 浙江大学

When evaluating the performance of data retrieval, it is important to measure the relevancy of the answer set.

☐ T ☒ F

答案正确: 1 分

 创建提问

在评估数据检索能力时，是要看已检索的在所有相关文档中的比例，与相关性无关。

3

1-3 分数 1

作者 沈鑫 单位 浙江大学

Precision is more important than recall when evaluating the explosive detection in airport security.

☐ T ☒ F

答案正确: 1 分

 创建提问

In airport, recall is more important than precision.

4.

1-4 分数 1

作者 杨欣豫 单位 浙江大学

While accessing a term by hashing in an inverted file index, range searches are expensive.

☐ T ☒ F

答案错误: 0 分

 创建提问

在倒排索引中使用哈希表检索一个词时，范围搜索的代价昂贵。

搜索树可以确定范围，但哈希表不能，存储不灵活。

5.

2-1 分数 2

作者 陈越 单位 浙江大学

When measuring the relevancy of the answer set, if the precision is high but the recall is low, it means that:

- ☐ A. most of the relevant documents are retrieved, but too many irrelevant documents are returned as well
- ☒ B. most of the retrieved documents are relevant, but still a lot of relevant documents are missed
- ☐ C. most of the relevant documents are retrieved, but the benchmark set is not large enough
- ☐ D. most of the retrieved documents are relevant, but the benchmark set is not large enough

答案正确: 2 分

 创建提问

已检索到的文档中相关文档较多，但仍然有许多文档没有检索到。

6.

2-2 分数 2

作者 陈越 单位 浙江大学

Which of the following is NOT concerned for measuring a search engine?

- ☐ A. How fast does it index
- ☒ B. How fast does it search
- ☐ C. How friendly is the interface
- ☐ D. How relevant is the answer set

答案正确: 2 分

💡 创建提问

7.

There are 28000 documents in the database. The statistic data for one query are shown in the following table. The recall is: __

	Relevant	Irrelevant
Retrieved	4000	12000
Not Retrieved	8000	4000

- ☐ A. 14%
- ☐ B. 25%
- ☒ C. 33%
- ☐ D. 50%

答案正确: 2 分

💡 创建提问

$$4000 / (4000 + 8000) = 33.3\%$$