Data Preparation and Analysis II

Dr. Anne Kayem

Hasso-Plattner-Institute
University of Potsdam

Indexing Documents

- Index - data structure for storing the descriptors

- Must be created for each document over all descriptors
  - Very large matrix, which normally does not fit into memory
  - Is only filled in a few places (sparse)

- More efficient memory structure needed
  - Inverted index

Word	Vector	#d1	#d2
Apple	e1=(1,0,0,0,0,0)	1	
Fruit	e2=(0,1,0,0,0,0)	1	2
Potato	e3=(0,0,1,0,0,0)		1
Onions	e4=(0,0,0,1,0,0)		1
Vegetable	e5=(0,0,0,0,1,0)	1	1

Example

- Text … „Hello everyone“, „This webpage ….“….“which webpage“…

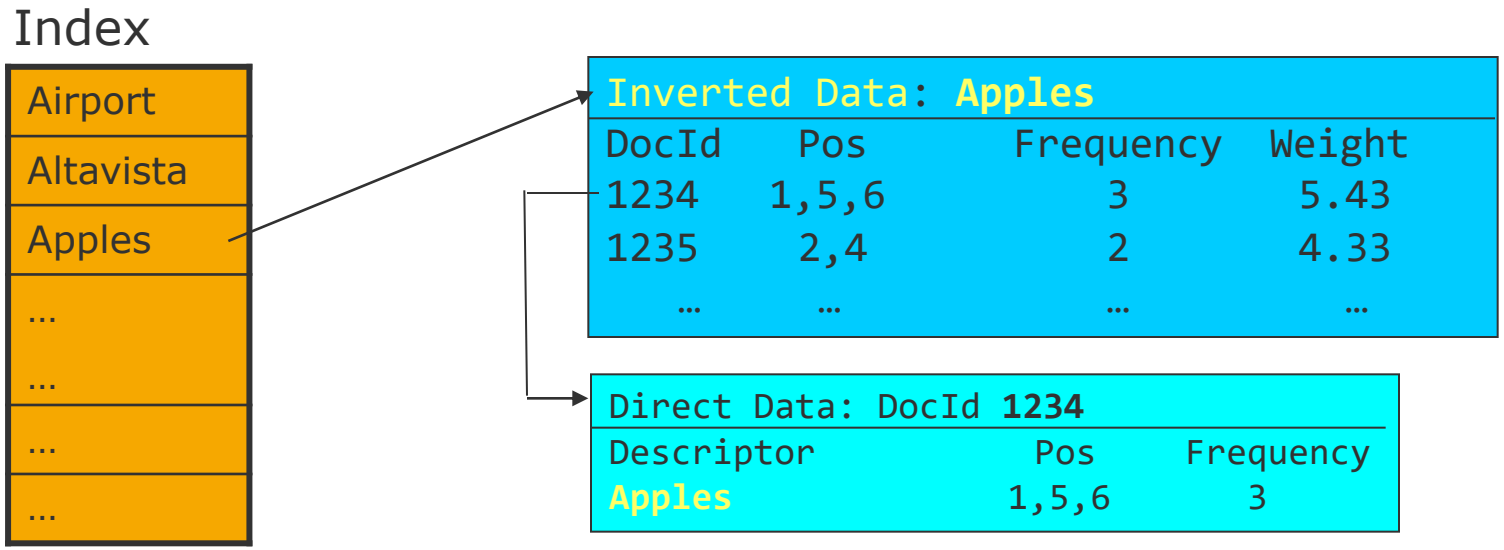
- Index with location in text can be as follows:

Text	Index
Hello	(1, 1)
everyone	(1, 2)
This	(2, 1)
webpage	(2, 2); (3, 2)

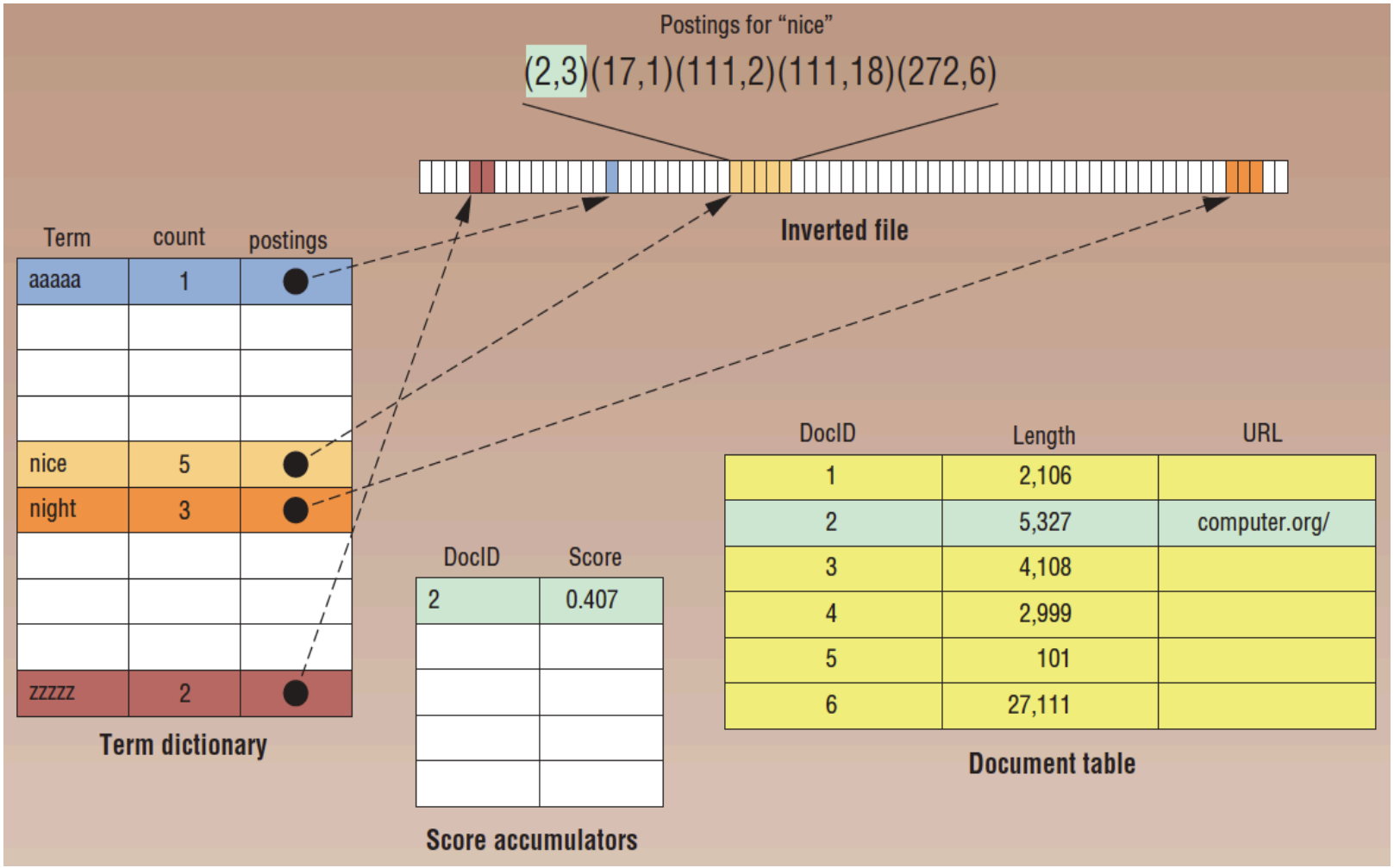
Inverted Index

- Need for fast response to search queries makes special data structures necessary

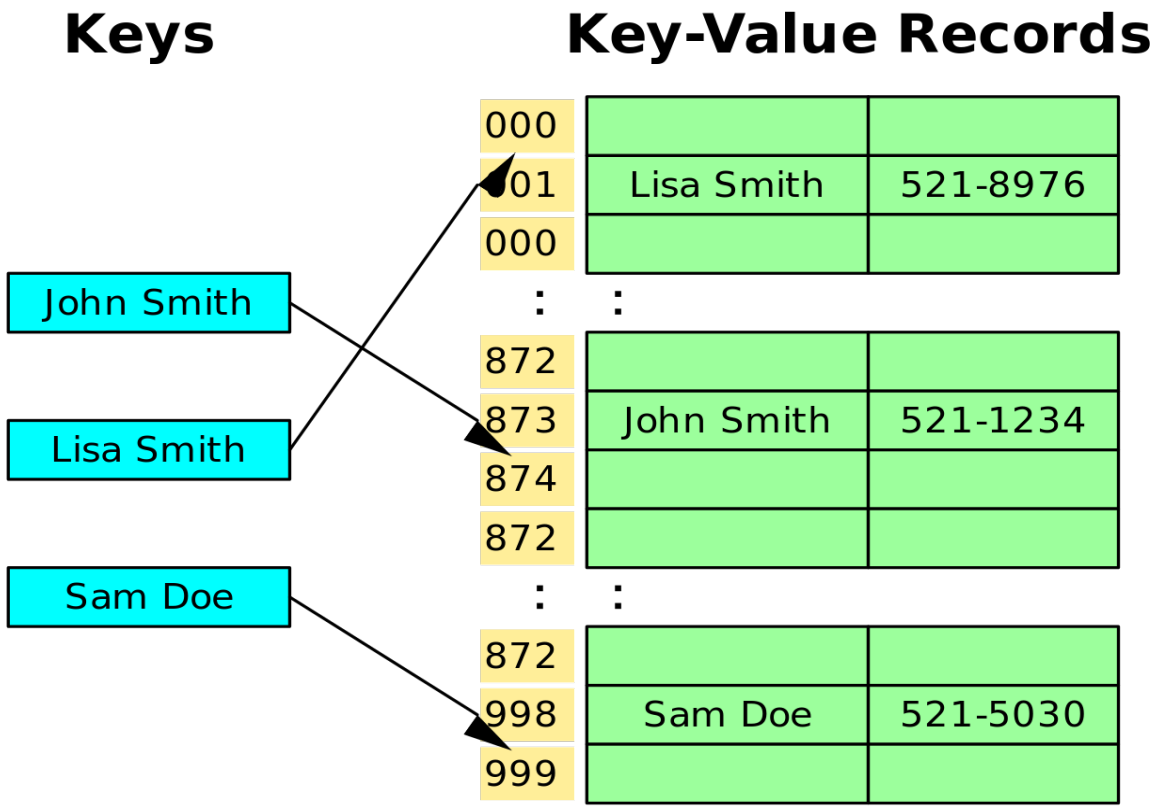
- Inverted file system - each descriptor is assigned a set of relevant documents



A Further Example



Example – Using Term Look Up



Scaling Up

Indexing is memory intensive

- Operations: sorting, storage, linking, searching, etc.
- **Example:** 500 terms in each of 20 billion pages → temporary file can contain 10 trillion entries.
- **A Possible Solution:**
  - **Document partitioning:** Divides up the URLs between machines in a cluster (similar to crawling strategy)
  - **Example:** Using 400 machines to index 20 billion pages (with even load per machine) → each machine handles 50 million pages.