Discussion Session
Week 10

INF 141: Information Retrieval
Winter 2009

Yasser Ganjisaffar

yganjisa@ics.uci.edu

Cosine Scoring

$$sim(q, d_i) = \frac{\vec{q} \cdot \vec{d}_i}{\left|\vec{q}\right|\left|\vec{d}_i\right|}$$

Scoring

$$sim(q, d_i) = \frac{\vec{q} \cdot \vec{d}_i}{\left|\vec{d}_i\right|}$$

Ranking (***q*** **vector is the same for all items and doesn't affect ranking).**

Efficient Cosine Ranking

- Find the *k* docs in the corpus "nearest" to the query $\Rightarrow$ *k* largest query-doc cosines.

- Efficient ranking:
  - Computing a single cosine efficiently.
  - Choosing the *k* largest cosine values efficiently.

Computing a single cosine

- For every term *i*, with each doc *j*, store term frequency tf_{ij}.

- At query time, accumulate:

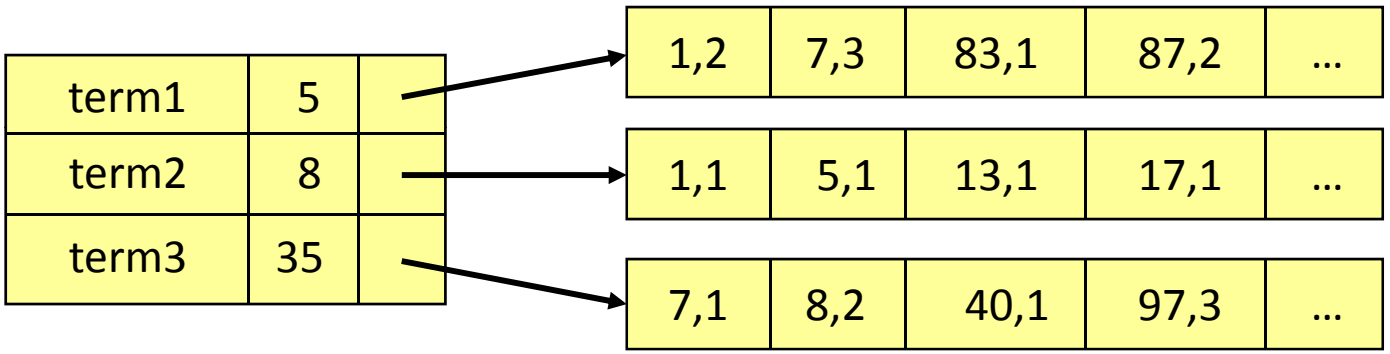
$$sim(\vec{q},\vec{d}_k) = \sum_{i=1}^{m} w_{i,q} \times w_{i,k}$$

Efficient Cosine Ranking

```
CosineScore(q)
 1  Initialize(Scores[d ∈ D])
 2  Initialize(Magnitude[d ∈ D])
 3  for each term(t ∈ q)
 4      do p ← FetchPostingsList(t)
 5          df_t ← GetCorpusWideStats(p)
 6          α_{t,q} ← WeightInQuery(t, q, df_t)
 7          for each {d, tf_{t,d}} ∈ p
 8              do Scores[d] += α_{t,q} · WeightInDocument(t, q, df_t)
 9  for d ∈ Scores
10      do Normalize(Scores[d], Magnitude[d])
11  return top K ∈ Scores
```

ignore

Posting Lists

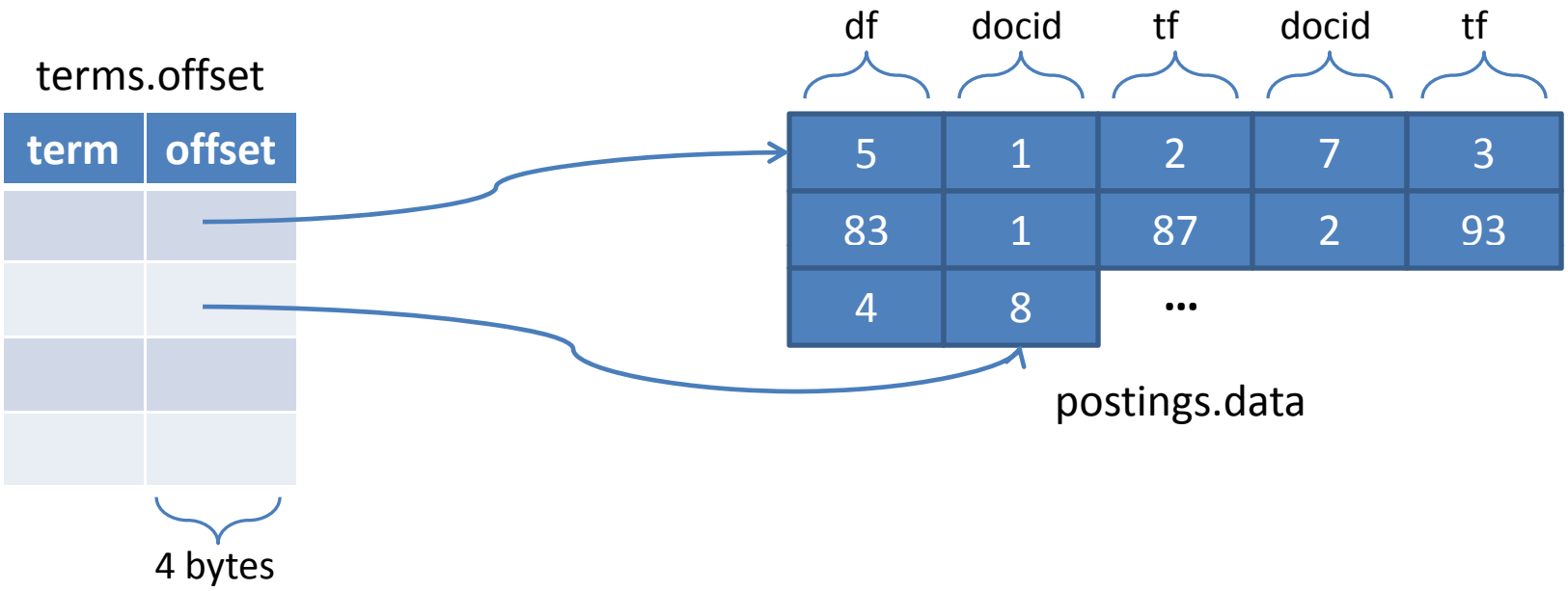


$$(1 + log(tf_{t,d})) * log\left(\frac{|corpus|}{df_t}\right)$$

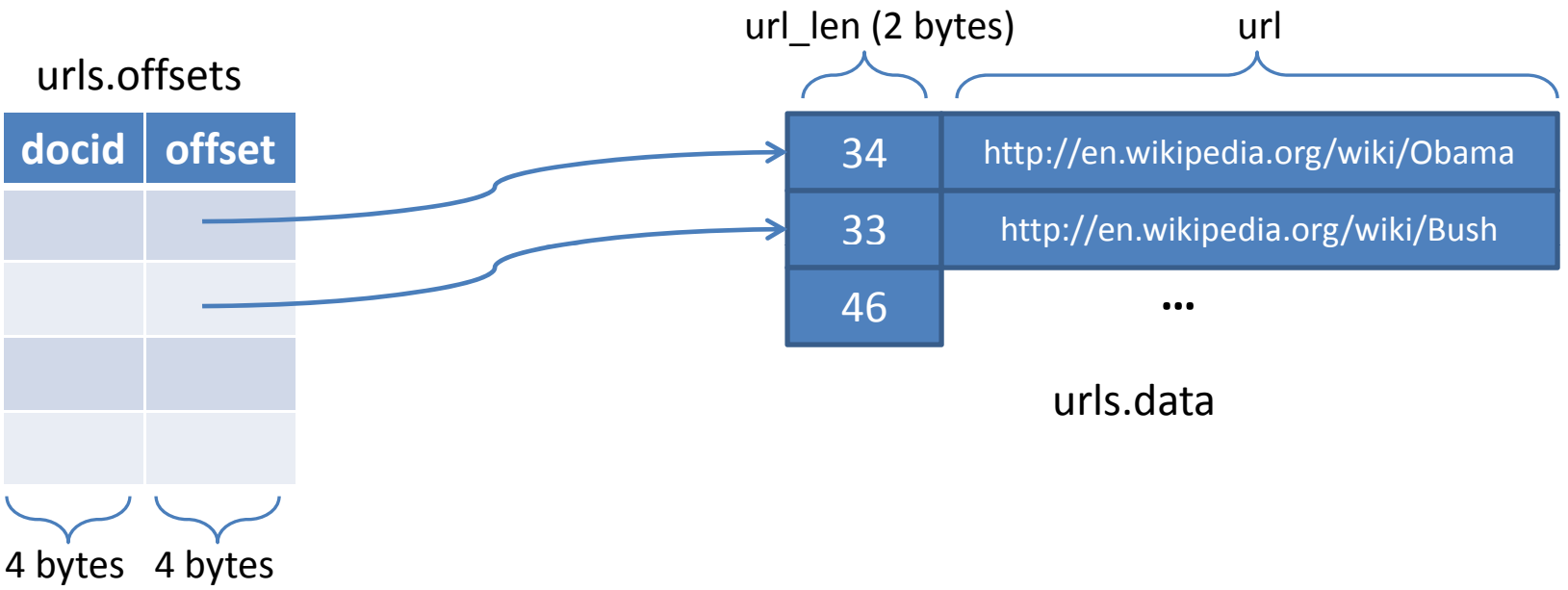
Java Primitive Data Types

Data Types	Size (byte)
byte	1
short	2
int	4
long	8
float	4
double	8
char	2

Suggested Data Structures



Suggested Data Structures (contd.)



Random Access Files

Initializations:

```
RandomAccessFile  urlsDataFile = new RandomAccessFile( "/path/urls.data", "rw");
RandomAccessFile  urlsOffsetFile = new RandomAccessFile( "/path/urls.offset", "rw");
int offset = 0;
```

writeURL(docid, url):

```
urlsOffsetFile.writeInt(docid);
urlsOffsetFile.writeInt(offset);
byte[] url_bytes = url.getBytes("UTF-8");
short url_len = (short) url_bytes.length;
urlsDataFile.writeShort(url_len);
urlsDataFile.write(url_bytes);
offset += 2 + url_bytes;
```

readURL(docid):

```
offset = getOffset(docid);
urlsDataFile.seek(offset);
short url_len = contentsFile.readShort();
byte[] url_bytes = new byte[url_len];
contentsFile.read(url_bytes);
return new String(url_bytes, "UTF-8");
```