Informatics 225
Computer Science 221

Information Retrieval

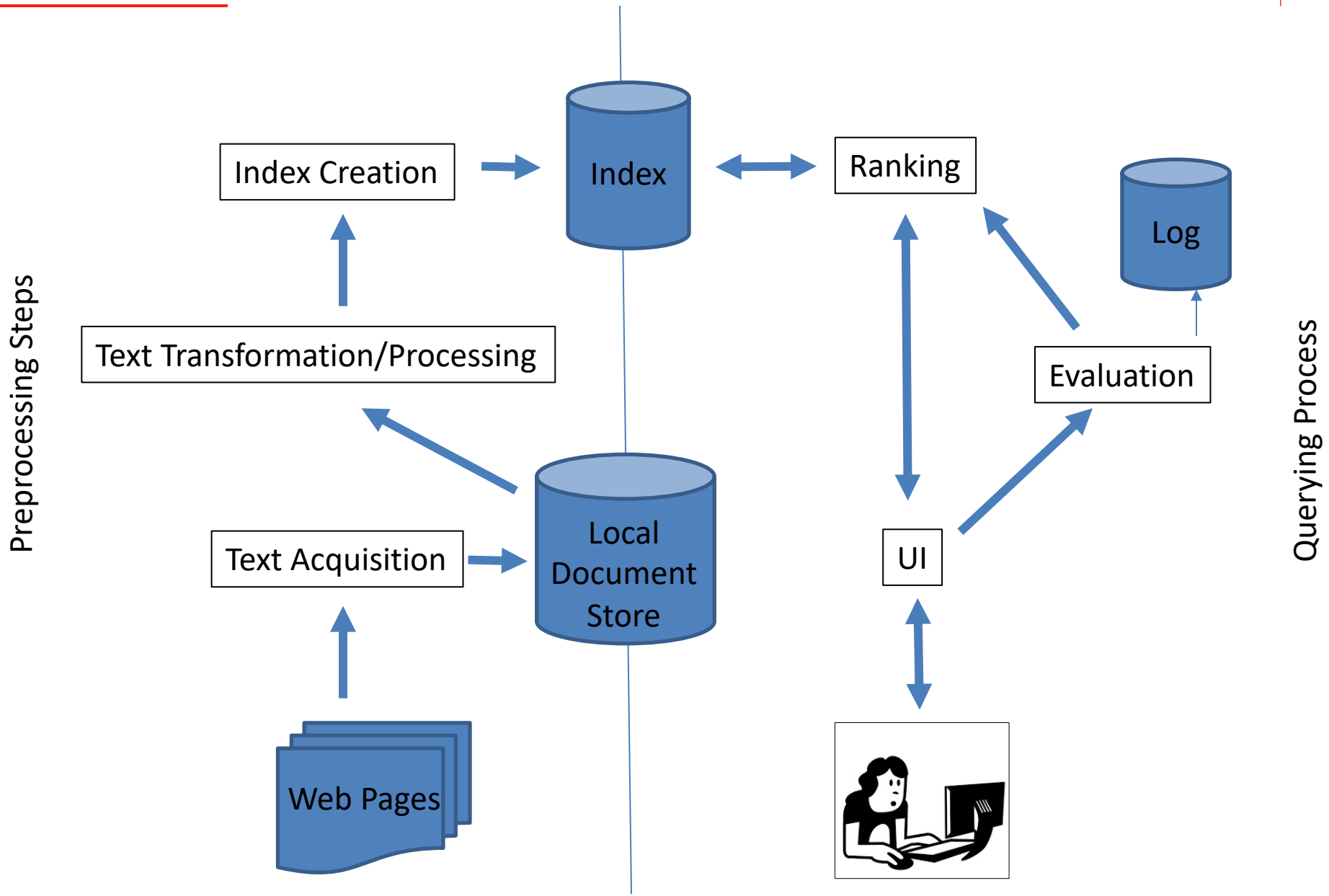
Lecture 11

These course materials borrow, with permission, from those of Prof. Cristina Videira Lopes, Prof. Alberto Krone-Martins, Addison Wesley 2008, Chris Manning, Pandu Nayak, Hinrich Schütze, Heike Adel, Sascha Rothe, Jerome H. Friedman, Robert Tibshirani, and Trevor Hastie. Powerpoint theme by Prof. André van der Hoek.

Storing documents

Information Retrieval

Architecture



Storing the Documents a web search engine

- After documents are converted you need to store them

- Many reasons to store converted document text
  - saves crawling time when page is not updated
  - provides efficient access to text for snippet generation, additional information extraction, etc.

Storing the Documents a web search engine

- After documents are converted you need to store them
- Many reasons to store converted document text
  - saves crawling time when page is not updated
  - provides efficient access to text for snippet generation, additional information extraction, etc.
- Database systems can provide document storage for **some** applications
  - **web search engines use customized document storage systems**

Storing the Documents

- Requirements for a search engine document storage system:
  - Random access
    - request the content of a document based on its URL
    - hash function based on URL is normally used
  - Adoption of compression and large files
    - reducing storage requirements and efficient access
  - Update capabilities
    - handling large volumes of new and modified documents
    - adding new anchor text

Large Files

- Webpages are small, so store many pages in a smaller number of large files, rather than each document in a file
  - avoids file I/O overhead
  - reduces seek time relative to read time

- Compound documents formats
  - used to store multiple documents in a file
  - e.g., TREC Web

TREC Web Format

```
<DOC>
<DOCNO>WTX001-B01-10</DOCNO>
<DOCHDR>
http://www.example.com/test.html 204.244.59.33 19970101013145 text/html 440
HTTP/1.0 200 OK
Date: Wed, 01 Jan 1997 01:21:13 GMT
Server: Apache/1.0.3
Content-type: text/html
Content-length: 270
Last-modified: Mon, 25 Nov 1996 05:31:24 GMT
</DOCHDR>
<HTML>
<TITLE>Tropical Fish Store</TITLE>
Coming soon!
</HTML>
</DOC>
<DOC>
<DOCNO>WTX001-B01-109</DOCNO>
<DOCHDR>
http://www.example.com/fish.html 204.244.59.33 19970101013149 text/html 440
HTTP/1.0 200 OK
Date: Wed, 01 Jan 1997 01:21:19 GMT
Server: Apache/1.0.3
Content-type: text/html
Content-length: 270
Last-modified: Mon, 25 Nov 1996 05:31:24 GMT
</DOCHDR>
<HTML>
<TITLE>Fish Information</TITLE>
This page will soon contain interesting
information about tropical fish.
</HTML>
</DOC>
```

TREC Web Format :: Document blocks

```
<DOC>
<DOCNO>WTX001-B01-10</DOCNO>
<DOCHDR>
http://www.example.com/test.html 204.244.59.33 19970101013145 text/html 440
HTTP/1.0 200 OK
Date: Wed, 01 Jan 1997 01:21:13 GMT
Server: Apache/1.0.3
Content-type: text/html
Content-length: 270
Last-modified: Mon, 25 Nov 1996 05:31:24 GMT
</DOCHDR>
<HTML>
<TITLE>Tropical Fish Store</TITLE>
Coming soon!
</HTML>
</DOC>
```

```
<DOC>
<DOCNO>WTX001-B01-109</DOCNO>
<DOCHDR>
http://www.example.com/fish.html 204.244.59.33 19970101013149 text/html 440
HTTP/1.0 200 OK
Date: Wed, 01 Jan 1997 01:21:19 GMT
Server: Apache/1.0.3
Content-type: text/html
Content-length: 270
Last-modified: Mon, 25 Nov 1996 05:31:24 GMT
</DOCHDR>
<HTML>
<TITLE>Fish Information</TITLE>
This page will soon contain interesting
information about tropical fish.
</HTML>
</DOC>
```

TREC Web Format :: Headers

```
<DOC>
<DOCNO>WTX001-B01-10</DOCNO>
<DOCHDR>
http://www.example.com/test.html 204.244.59.33 19970101013145 text/html 440
HTTP/1.0 200 OK
Date: Wed, 01 Jan 1997 01:21:13 GMT
Server: Apache/1.0.3
Content-type: text/html
Content-length: 270
Last-modified: Mon, 25 Nov 1996 05:31:24 GMT
</DOCHDR>
<HTML>
<TITLE>Tropical Fish Store</TITLE>
Coming soon!
</HTML>
</DOC>
<DOC>
<DOCNO>WTX001-B01-109</DOCNO>
<DOCHDR>
http://www.example.com/fish.html 204.244.59.33 19970101013149 text/html 440
HTTP/1.0 200 OK
Date: Wed, 01 Jan 1997 01:21:19 GMT
Server: Apache/1.0.3
Content-type: text/html
Content-length: 270
Last-modified: Mon, 25 Nov 1996 05:31:24 GMT
</DOCHDR>
<HTML>
<TITLE>Fish Information</TITLE>
This page will soon contain interesting
information about tropical fish.
</HTML>
</DOC>
```

TREC Web Format :: Content

```
<DOC>
<DOCNO>WTX001-B01-10</DOCNO>
<DOCHDR>
http://www.example.com/test.html 204.244.59.33 19970101013145 text/html 440
HTTP/1.0 200 OK
Date: Wed, 01 Jan 1997 01:21:13 GMT
Server: Apache/1.0.3
Content-type: text/html
Content-length: 270
Last-modified: Mon, 25 Nov 1996 05:31:24 GMT
</DOCHDR>
<HTML>
<TITLE>Tropical Fish Store</TITLE>
Coming soon!
</HTML>
</DOC>
<DOC>
<DOCNO>WTX001-B01-109</DOCNO>
<DOCHDR>
http://www.example.com/fish.html 204.244.59.33 19970101013149 text/html 440
HTTP/1.0 200 OK
Date: Wed, 01 Jan 1997 01:21:19 GMT
Server: Apache/1.0.3
Content-type: text/html
Content-length: 270
Last-modified: Mon, 25 Nov 1996 05:31:24 GMT
</DOCHDR>
<HTML>
<TITLE>Fish Information</TITLE>
This page will soon contain interesting
information about tropical fish.
</HTML>
</DOC>
```

TREC Web Format :: Header document identifier

```
<DOC>
<DOCNO>WTX001-B01-10</DOCNO>
<DOCHDR>
http://www.example.com/test.html 204.244.59.33 19970101013145 text/html 440
HTTP/1.0 200 OK
Date: Wed, 01 Jan 1997 01:21:13 GMT
Server: Apache/1.0.3
Content-type: text/html
Content-length: 270
Last-modified: Mon, 25 Nov 1996 05:31:24 GMT
</DOCHDR>
<HTML>
<TITLE>Tropical Fish Store</TITLE>
Coming soon!
</HTML>
</DOC>
<DOC>
<DOCNO>WTX001-B01-109</DOCNO>
<DOCHDR>
http://www.example.com/fish.html 204.244.59.33 19970101013149 text/html 440
HTTP/1.0 200 OK
Date: Wed, 01 Jan 1997 01:21:19 GMT
Server: Apache/1.0.3
Content-type: text/html
Content-length: 270
Last-modified: Mon, 25 Nov 1996 05:31:24 GMT
</DOCHDR>
<HTML>
<TITLE>Fish Information</TITLE>
This page will soon contain interesting
information about tropical fish.
</HTML>
</DOC>
```

TREC Web Format :: Header modification tag

```
<DOC>
<DOCNO>WTX001-B01-10</DOCNO>
<DOCHDR>
http://www.example.com/test.html 204.244.59.33 19970101013145 text/html 440
HTTP/1.0 200 OK
Date: Wed, 01 Jan 1997 01:21:13 GMT
Server: Apache/1.0.3
Content-type: text/html
Content-length: 270
Last-modified: Mon, 25 Nov 1996 05:31:24 GMT
</DOCHDR>
<HTML>
<TITLE>Tropical Fish Store</TITLE>
Coming soon!
</HTML>
</DOC>
<DOC>
<DOCNO>WTX001-B01-109</DOCNO>
<DOCHDR>
http://www.example.com/fish.html 204.244.59.33 19970101013149 text/html 440
HTTP/1.0 200 OK
Date: Wed, 01 Jan 1997 01:21:19 GMT
Server: Apache/1.0.3
Content-type: text/html
Content-length: 270
Last-modified: Mon, 25 Nov 1996 05:31:24 GMT
</DOCHDR>
<HTML>
<TITLE>Fish Information</TITLE>
This page will soon contain interesting
information about tropical fish.
</HTML>
</DOC>
```

Compression

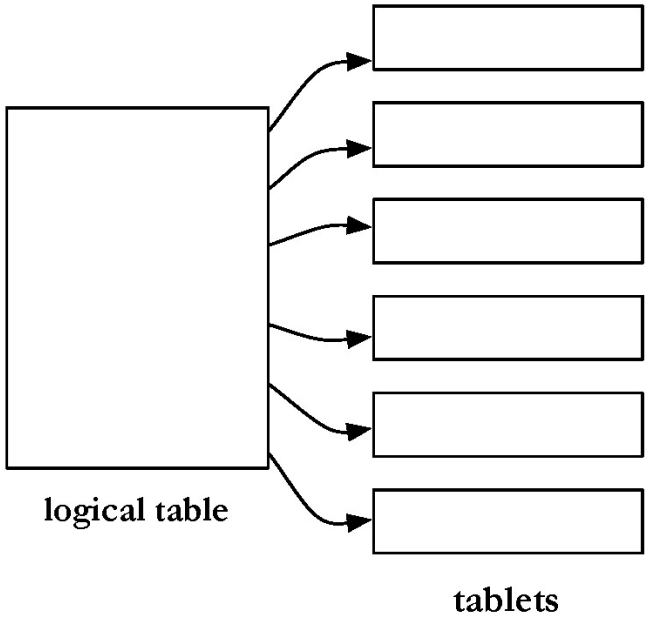
- Text is highly redundant (or predictable)

Compression

- Text is highly redundant (or predictable)
- Lossless compression techniques exploit this redundancy to make files smaller without losing any of the content
  - One can also perform compression of indexes
- Popular algorithms can compress HTML and XML text by 80%
  - e.g., DEFLATE (zip, gzip) and LZW (UNIX compress, PDF)
  - may compress large files in blocks to make access faster

BigTable

- Google's document storage system
  - Customized for *storing*, *finding*, and *updating* web pages
  - Handles large collection sizes using *inexpensive computers*

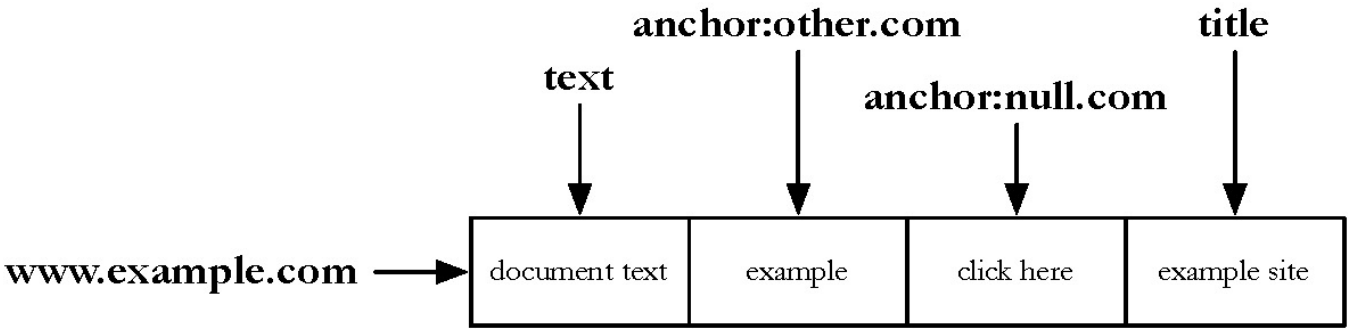


BigTable advantages and drawbacks

- No query language: no complex queries to optimize
- Only row-level transactions
- Tablets are stored in a replicated file system that is accessible by all BigTable servers
- Any changes to a BigTable tablet are recorded to a transaction log, which is also stored in a shared file system
- If any tablet server crashes, another server can read the tablet data and transaction log from the file system and take over
- Tablets may be compressed

BigTable organization

- Logically organized into rows
- A row stores data for a single web page
- Columns can have multiple timestamps



- Combination of a row key, a column key, and a timestamp point to a single *cell* in the row

BigTable organization

- BigTable can have a huge number of columns per row
  - all rows have the same column groups
  - not all rows have the same columns
  - important for reducing disk reads to access document data

- Rows are partitioned into tablets based on their row keys
  - E.g. URLs starting with a certain letter can be in a certain tablet.
  - simplifies determining which server is appropriate to access

Detecting duplicates and removing noise

Information Retrieval

Detecting Duplicates

- Duplicate and near-duplicate documents occur in many situations
  - Copies, versions, plagiarism, spam, mirror sites
  - 30% of the web pages in a large crawl are exact or near duplicates of the rest (Fetterly et al., 2003)

- Duplicates consume significant resources during crawling, indexing, and search
  - Add little value to most users *(buy may add value to some use cases)*

Duplicate Detection

- *Exact* duplicate detection is relatively easy

- *Checksum* techniques
  - A checksum is a value computed from the content of the document
    - e.g., sum of the bytes in the document file

T	r	o	p	i	c	a	l		f	i	s	h	*Sum*
54	72	6F	70	69	63	61	6C	20	66	69	73	68	508

Duplicate Detection

- *Exact* duplicate detection is relatively easy

- *Checksum* techniques
  - A checksum is a value computed from the content of the document
    - e.g., sum of the bytes in the document file

T	r	o	p	i	c	a	l		f	i	s	h	*Sum*
54	72	6F	70	69	63	61	6C	20	66	69	73	68	508

  - Possible for files with different text to have same checksum

Duplicate Detection

- *Exact* duplicate detection is relatively easy
- *Checksum* techniques
  - A checksum is a value computed from the content of the document
    - e.g., sum of the bytes in the document file

T	r	o	p	i	c	a	l		f	i	s	h	*Sum*
54	72	6F	70	69	63	61	6C	20	66	69	73	68	508

  - Possible for files with different text to have same checksum
- Functions such as a *cyclic redundancy check* (CRC), have been developed that consider the positions of the bytes
  - Still prone to collisions, but very rare
  - Other functions as BLAKE2/3, MD5, SHA1, SHA2, etc…
- Need to be fast

Near-Duplicate Detection

- Near duplicate is a more challenging task
  - Are web pages with same text content but different advertising or format near-duplicates?

Near-Duplicate Detection

- Near duplicate is a more challenging task
  - Are web pages with same text content but different advertising or format near-duplicates?

- A near-duplicate document is defined using a threshold value for some similarity measure between pairs of documents
  - e.g., document *D1* is a near-duplicate of document *D2* if more than 90% of the words in the documents are the same

Near-Duplicate Detection

- Near duplicate is a more challenging task
  - Are web pages with same text content but different advertising or format near-duplicates?
- A near-duplicate document is defined using a threshold value for some similarity measure between pairs of documents
  - e.g., document *D1* is a near-duplicate of document *D2* if more than 90% of the words in the documents are the same
- *Do you see how you can use Assignment 1 here?*

Near-Duplicate Detection

- *Two scenarios:*
  - *Search*:
    - find near-duplicates of a document *D*
    - *O(N)* comparisons
  - *Discovery*:
    - find all pairs of near-duplicate documents in the collection
    - $O(N^2)$ comparisons

Near-Duplicate Detection

- *Two scenarios:*
  - *Search*:
    - find near-duplicates of a document *D*
    - *O(N)* comparisons
  - *Discovery*:
    - find all pairs of near-duplicate documents in the collection
    - $O(N^2)$ comparisons

- Normal IR techniques are effective for search scenario

- For discovery, other techniques used to generate compact representations known as fingerprints

Fingerprints

1. The document is parsed into words. Non-word content, such as punctuation, HTML tags, and additional whitespace, is removed.

2. The words are grouped into contiguous *n-grams* for some n. These are usually overlapping sequences of words, although some techniques use non-overlapping sequences.

3. Some of the n-grams are selected to represent the document.

4. The selected n-grams are hashed to improve retrieval efficiency and further reduce the size of the representation.

5. The hash values are stored, typically in an inverted index.

6. Documents are compared using overlap of fingerprints

Fingerprint Example

Tropical fish include fish found in tropical environments around the world, including both freshwater and salt water species.

(a) Original text

Fingerprint Example

Tropical fish include fish found in tropical environments around the world, including both freshwater and salt water species.

(a) Original text

tropical fish include, fish include fish, include fish found, fish found in, found in tropical, in tropical environments, tropical environments around, environments around the, around the world, the world including, world including both, including both freshwater, both freshwater and, freshwater and salt, and salt water, salt water species

(b) 3-grams

Fingerprint Example

Tropical fish include fish found in tropical environments around the world, including both freshwater and salt water species.

(a) Original text

tropical fish include, fish include fish, include fish found, fish found in, found in tropical, in tropical environments, tropical environments around, environments around the, around the world, the world including, world including both, including both freshwater, both freshwater and, freshwater and salt, and salt water, salt water species

(b) 3-grams

938 664 463 822 492 798 78 969 143 236 913 908 694 553 870 779

(c) Hash values

Fingerprint Example

Tropical fish include fish found in tropical environments around the world, including both freshwater and salt water species.

(a) Original text

tropical fish include, fish include fish, include fish found, fish found in, found in tropical, in tropical environments, tropical environments around, environments around the, around the world, the world including, world including both, including both freshwater, both freshwater and, freshwater and salt, and salt water, salt water species

(b) 3-grams

938 664 463 822 492 798 78 969 143 236 913 908 694 553 870 779

(c) Hash values

664 492 236 908

(d) Selected hash values using H mod 4 = 0

Fingerprint Example

- So, how similar are two text files A and B?

Fingerprint Example

- So, how similar are two text files A and B?
  - Compute the fingerprints F_A and F_B of each text.

Fingerprint Example

- So, how similar are two text files A and B?
  - Compute the fingerprints F_A and F_B of each text.
  - The similarity $S_{A,B}$ between A and B is simply the fraction of the intersection over the union of the fingerprint sets of A and B.

$$S_{A,B} = \frac{\mathrm{card}(\mathcal{F}_A \cap \mathcal{F}_B)}{\mathrm{card}(\mathcal{F}_A \cup \mathcal{F}_B)}$$

Fingerprint Example

- So, how similar are two text files A and B?
  - Compute the fingerprints F_A and F_B of each text.
  - The similarity $S_{A,B}$ between A and B is simply the fraction of the intersection over the union of the fingerprint sets of A and B.
- Are text A and text B near duplicates?
  - Define a threshold level τ.
  - If similarity is greater or equal than the threshold, they are near duplicates **under your definition of threshold.**

$$S_{A,B} \geq \tau \implies \text{A \& B are near duplicates}$$