Topic: Duplicate Detection and Similarity Computing

UCSB 290N, 2015

Tao Yang

Some of slides are from text book [CMS] and Rajaraman/Ullman

Table of Content

- Motivation
- Shingling for duplicate comparison
- Minhashing
- LSH

Applications of Duplicate Detection and Similarity Computing

- **Duplicate and near-duplicate documents occur in many situations**
  - Copies, versions, plagiarism, spam, mirror sites
  - 30-60+% of the web pages in a large crawl can be exact or near duplicates of pages in the other 70%
  - Duplicates consume significant resources during crawling, indexing, and search
- **Similar query suggestions**
- **Advertisement: coalition and spam detection**
- **Product recommendation based on similar product features or user interests**

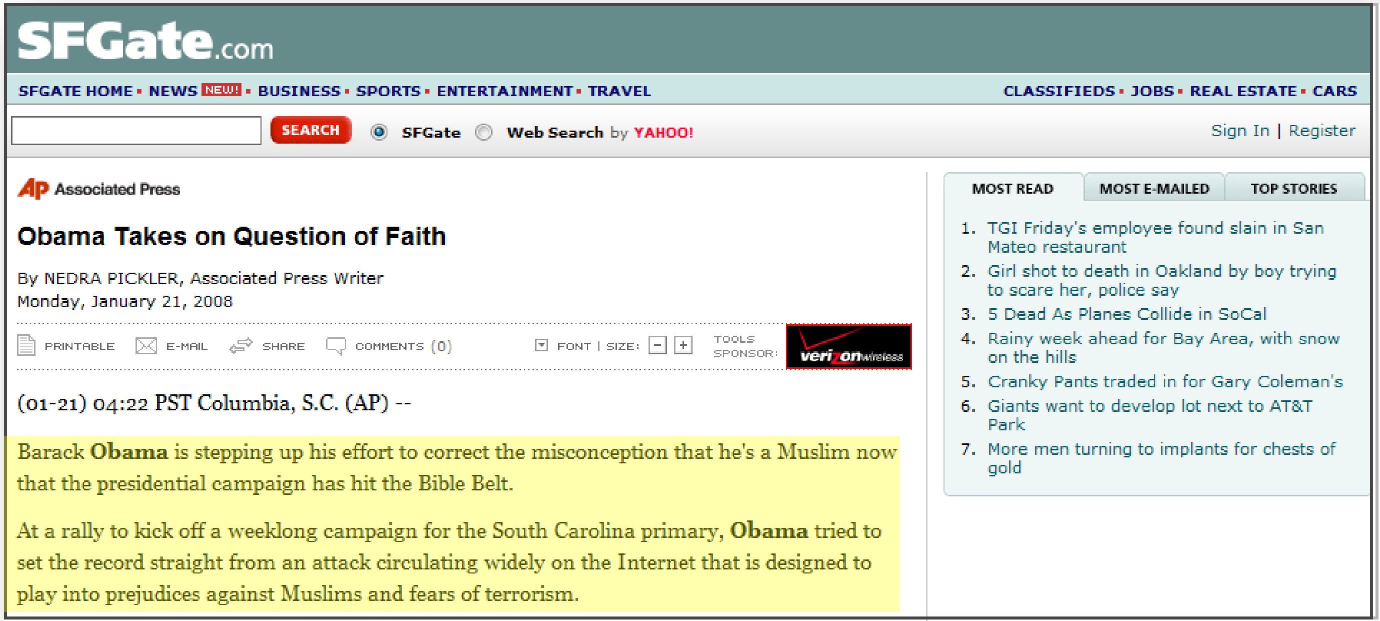
Duplicate Detection

- ***Exact* duplicate detection is relatively easy**
  - Content fingerprints
  - MD5, *cyclic redundancy check* (CRC)
- ***Checksum* techniques**
  - A checksum is a value that is computed based on 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

Near-Duplicate News Articles



SFGate.com

SFGATE HOME · NEWS NEW! · BUSINESS · SPORTS · ENTERTAINMENT · TRAVEL   CLASSIFIEDS · JOBS · REAL ESTATE · CARS

SEARCH ● SFGate ○ Web Search by YAHOO!   Sign In | Register

AP Associated Press

Obama Takes on Question of Faith

By NEDRA PICKLER, Associated Press Writer
Monday, January 21, 2008

PRINTABLE E-MAIL SHARE COMMENTS (0) FONT | SIZE: − + TOOLS SPONSOR: verizon wireless

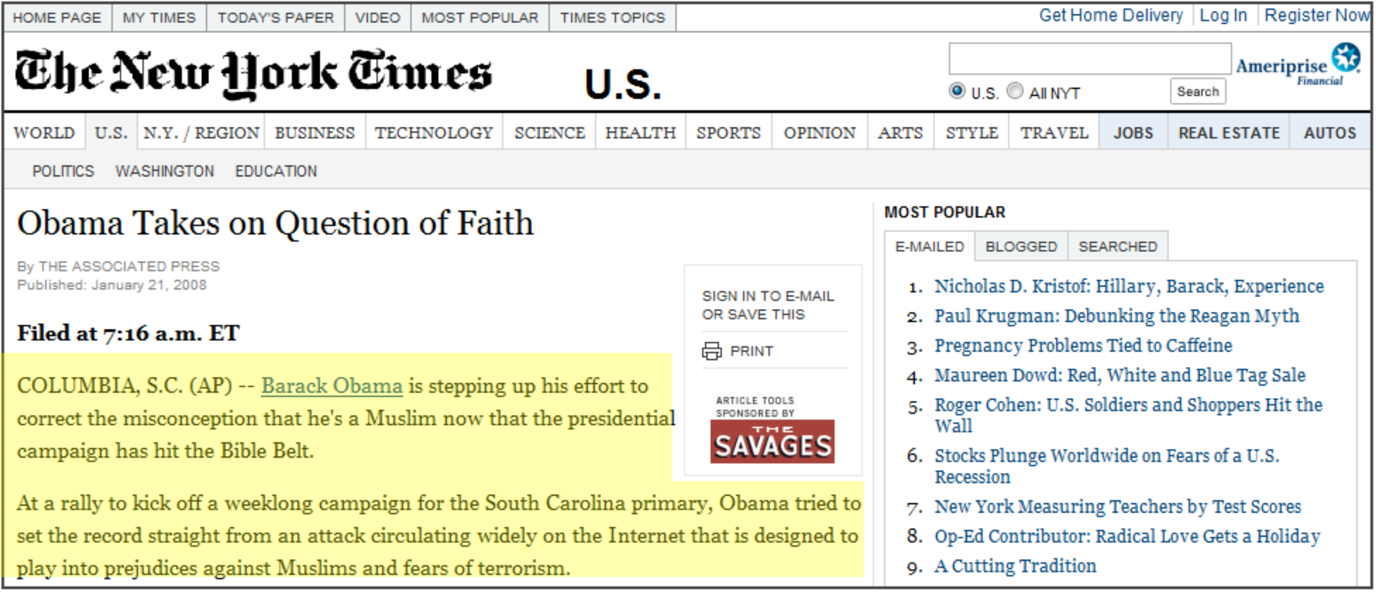
(01-21) 04:22 PST Columbia, S.C. (AP) --

Barack **Obama** is stepping up his effort to correct the misconception that he's a Muslim now that the presidential campaign has hit the Bible Belt.

At a rally to kick off a weeklong campaign for the South Carolina primary, **Obama** tried to set the record straight from an attack circulating widely on the Internet that is designed to play into prejudices against Muslims and fears of terrorism.

MOST READ | MOST E-MAILED | TOP STORIES

1. TGI Friday's employee found slain in San Mateo restaurant
2. Girl shot to death in Oakland by boy trying to scare her, police say
3. 5 Dead As Planes Collide in SoCal
4. Rainy week ahead for Bay Area, with snow on the hills
5. Cranky Pants traded in for Gary Coleman's
6. Giants want to develop lot next to AT&T Park
7. More men turning to implants for chests of gold



HOME PAGE | MY TIMES | TODAY'S PAPER | VIDEO | MOST POPULAR | TIMES TOPICS   Get Home Delivery | Log In | Register Now

The New York Times   U.S.

● U.S. ○ All NYT Search Ameriprise Financial

WORLD | U.S. | N.Y. / REGION | BUSINESS | TECHNOLOGY | SCIENCE | HEALTH | SPORTS | OPINION | ARTS | STYLE | TRAVEL | JOBS | REAL ESTATE | AUTOS

POLITICS WASHINGTON EDUCATION

Obama Takes on Question of Faith

By THE ASSOCIATED PRESS
Published: January 21, 2008

SIGN IN TO E-MAIL OR SAVE THIS

PRINT

ARTICLE TOOLS SPONSORED BY THE SAVAGES

Filed at 7:16 a.m. ET

COLUMBIA, S.C. (AP) -- Barack Obama is stepping up his effort to correct the misconception that he's a Muslim now that the presidential campaign has hit the Bible Belt.

At a rally to kick off a weeklong campaign for the South Carolina primary, Obama tried to set the record straight from an attack circulating widely on the Internet that is designed to play into prejudices against Muslims and fears of terrorism.

MOST POPULAR

E-MAILED | BLOGGED | SEARCHED

1. Nicholas D. Kristof: Hillary, Barack, Experience
2. Paul Krugman: Debunking the Reagan Myth
3. Pregnancy Problems Tied to Caffeine
4. Maureen Dowd: Red, White and Blue Tag Sale
5. Roger Cohen: U.S. Soldiers and Shoppers Hit the Wall
6. Stocks Plunge Worldwide on Fears of a U.S. Recession
7. New York Measuring Teachers by Test Scores
8. Op-Ed Contributor: Radical Love Gets a Holiday
9. A Cutting Tradition

Near-Duplicate Detection

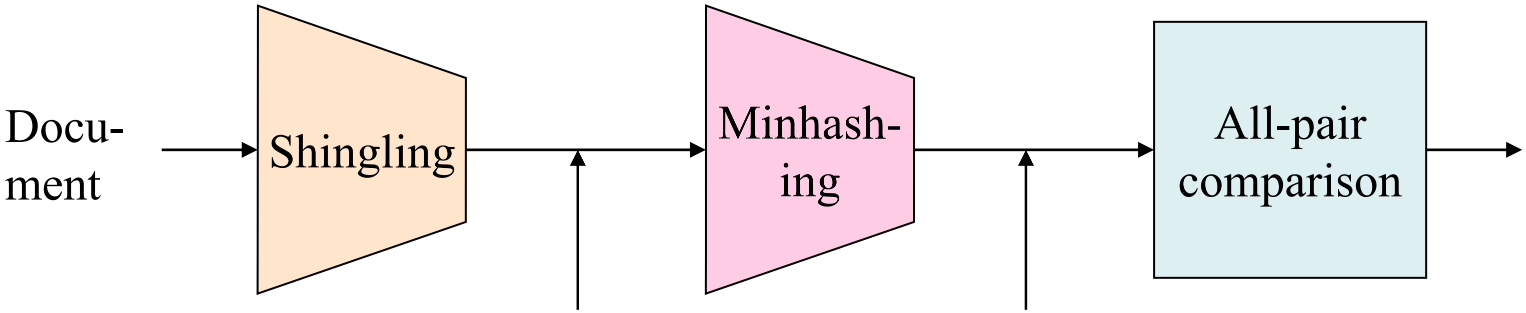
- **More challenging task**
  - Are web pages with same text context but different advertising or format near-duplicates?
- ***Near-Duplication*: Approximate match**
  - Compute syntactic similarity with an edit-distance measure
  - Use similarity threshold to detect near-duplicates
    - E.g., Similarity > 80% => Documents are “near duplicates”
    - Not transitive though sometimes used transitively

Near-Duplicate Detection

- ***Search***:
  - find near-duplicates of a document *D*
  - *O(N)* comparisons required
- ***Discovery***:
  - find all pairs of near-duplicate documents in the collection
  - $O(N^2)$ comparisons
- **IR techniques are effective for search scenario**
- **For discovery, other techniques used to generate compact representations**

Two Techniques for Computing Similarity

1. ***Shingling*** **: convert documents, emails, etc., to fingerprint sets.**
2. ***Minhashing*** **: convert large sets to short signatures, while preserving similarity.**



Document

Shingling

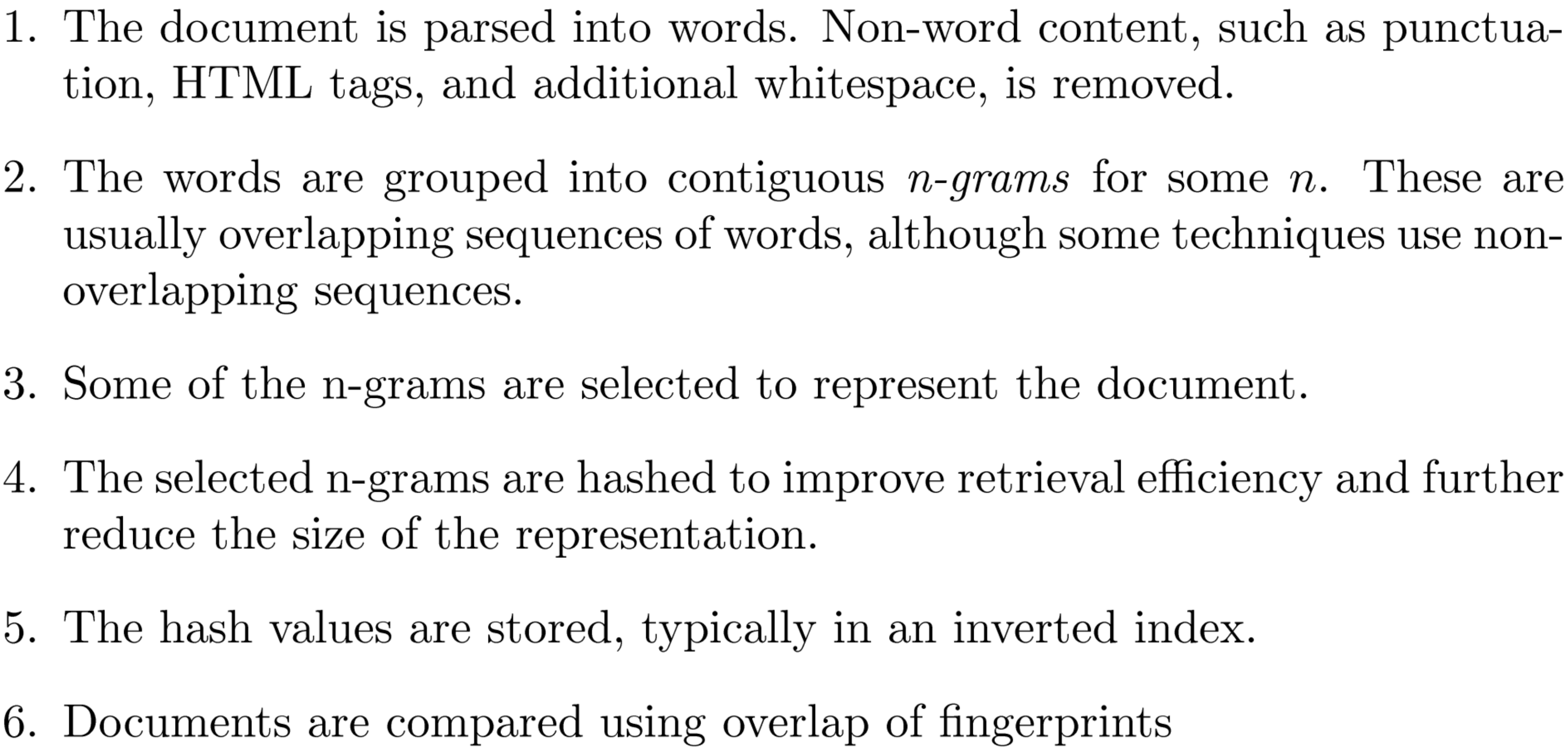
Minhashing

All-pair comparison

The set of strings of length *k* that appear in the document

Signatures : short integer vectors that represent the sets, and reflect their similarity

Fingerprint Generation Process for Web Documents



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

Computing Similarity with Shingles

- **Shingles (Word *k*-Grams) [Brin95, Brod98]**

  "a rose is a rose is a rose" =>

  a_rose_is_a

  rose_is_a_rose

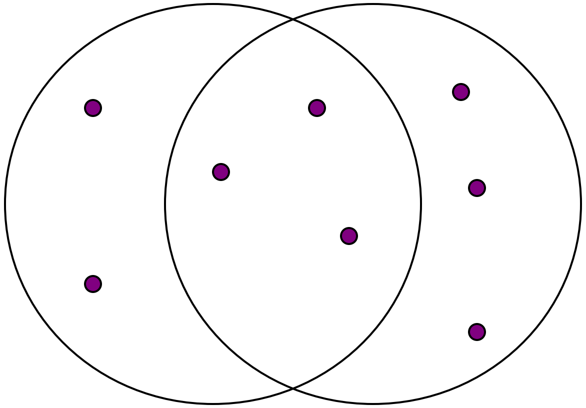
  is_a_rose_is

- **Similarity Measure between two docs (= sets of shingles)**
  - Size_of_Intersection / Size_of_Union

Jaccard measure

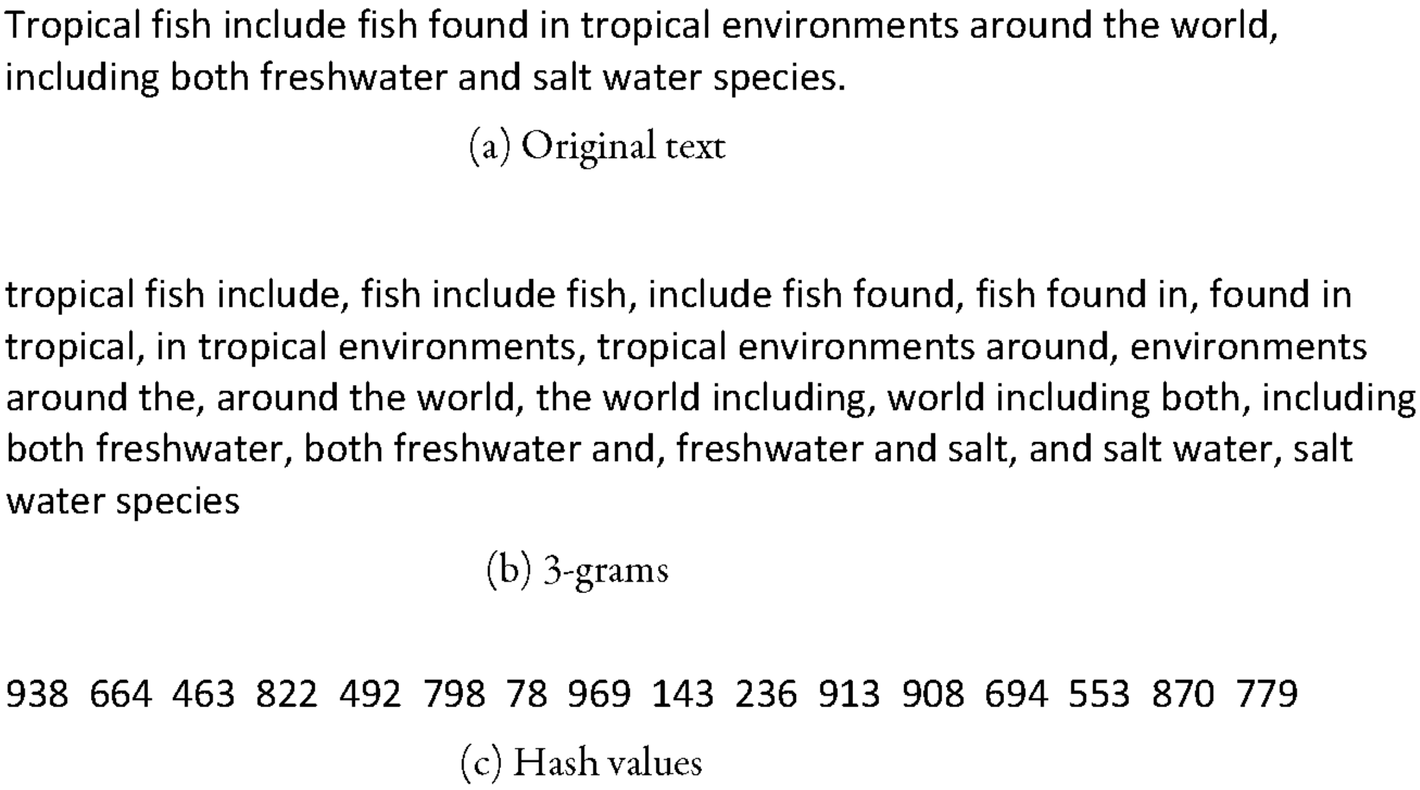
Example: Jaccard Similarity

- **The *Jaccard similarity* of two sets is the size of their intersection divided by the size of their union.**
  - *Sim* $(C_1, C_2) = |C_1 \cap C_2| / |C_1 \cup C_2|$.



3 in intersection.
8 in union.
Jaccard similarity
= 3/8

Fingerprint Example for Web Documents



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

Approximated Representation with Sketching

- **Computing <u>exact</u> set intersection of shingles between all pairs of documents is expensive**
  - Approximate using a subset of shingles (called sketch vectors)
  - Create a sketch vector using minhashing.
    - For doc *d*, sketch$_d$[i] is computed as follows:
    - Let *f* map all shingles in the universe to $0..2^m$
    - Let π_i be a specific random permutation on $0..2^m$
    - Pick MIN $\pi_i(f(s))$ over all shingles *s* in this document *d*
  - Documents which share more than t (say 80%) in sketch vector's elements are similar

Example: Min-hash

Round 1:

ordering = [cat, dog, mouse, banana]

Document 1:
{mouse, dog}
MH-signature = dog

Document 2:
{cat, mouse}
MH-signature = cat

Example: Min-hash

Round 2:

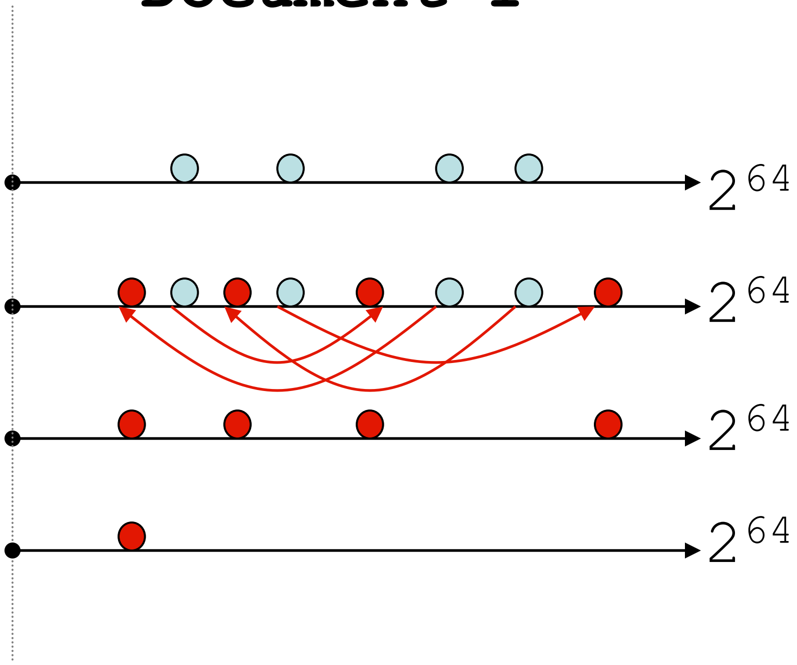
ordering = [banana, mouse, cat, dog]

Document 1:
{mouse, dog}
MH-signature = mouse

Document 2:
{cat, mouse}
MH-signature = mouse

Computing Sketch[i] for Doc1

Document 1

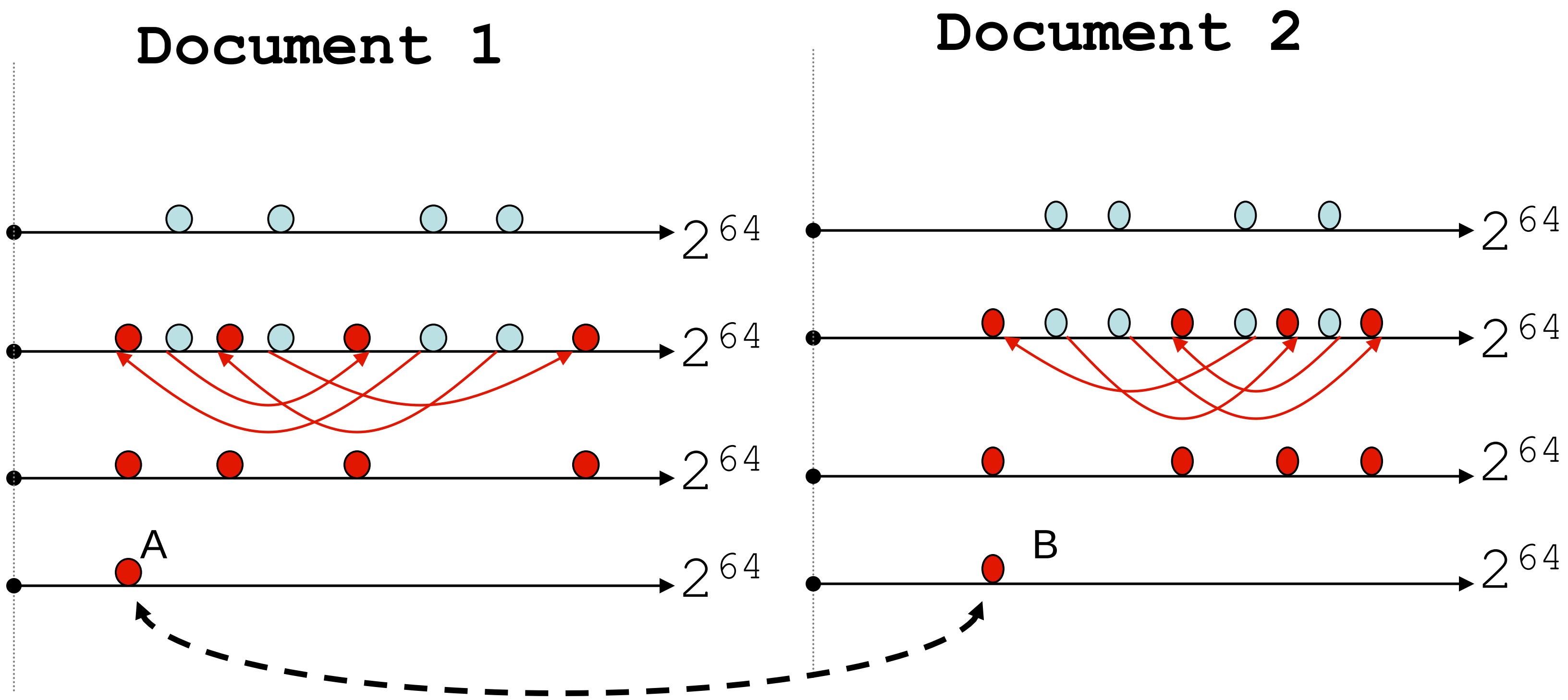


Start with 64 bit shingles

Permute on the number line with π_i

Pick the min value

Test if Doc1.Sketch[i] = Doc2.Sketch[i]



Are these equal?

Test for 200 random permutations: $\pi_1, \pi_2, \ldots \pi_{200}$

Shingling with minhashing

- **Given two documents d1, d2.**
- **Let S1 and S2 be their shingle sets**
- **Resemblance = |Intersection of S1 and S2| / | Union of S1 and S2|.**
- **Let Alpha = min (π (S1))**
- **Let Beta = min (π(S2))**
  - Probability (Alpha = Beta) = Resemblance
  - Computing this by sampling (e.g. 200 times).

Proof with Boolean Matrices

- **Rows = elements of the universal set.**
- **Columns = sets.**
- **1 in row *e* and column *S* if and only if *e* is a member of *S*.**
- **Column similarity is the Jaccard similarity of the sets of their rows with 1.**
- **Typical matrix is sparse.**

$$\mathrm{sim}_J(C_i, C_j) = \frac{|C_i \cap C_j|}{|C_i \cup C_j|}$$

C_1	C_2		
0	1		*
1	0		*
1	1	*	*
0	0		
1	1	*	*
0	1		*

Sim (C_1, C_2) =

2/5 = 0.4

Key Observation

- For columns C_i, C_j, four types of rows

	C_i	C_j
A	1	1
B	1	0
C	0	1
D	0	0

- Overload notation: A = # of rows of type A
- Claim

$$\mathrm{sim}_J(C_i, C_j) = \frac{A}{A + B + C}$$

Minhashing

- Imagine the rows permuted randomly.
- “hash” function *h* (*C*) = the index of the first (in the permuted order) row with 1 in column *C*.
- Use several (e.g., 100) independent hash functions to create a signature.
- The *similarity of signatures* is the fraction of the hash functions in which they agree.

Property

- **The probability (over all permutations of the rows) that *h* (C_1) = *h* (C_2) is the same as *Sim* (C_1, C_2).**

$$P\left[\, h(C_i) = h(C_j) \,\right] = \mathrm{sim}_J\left(C_i, C_j\right)$$

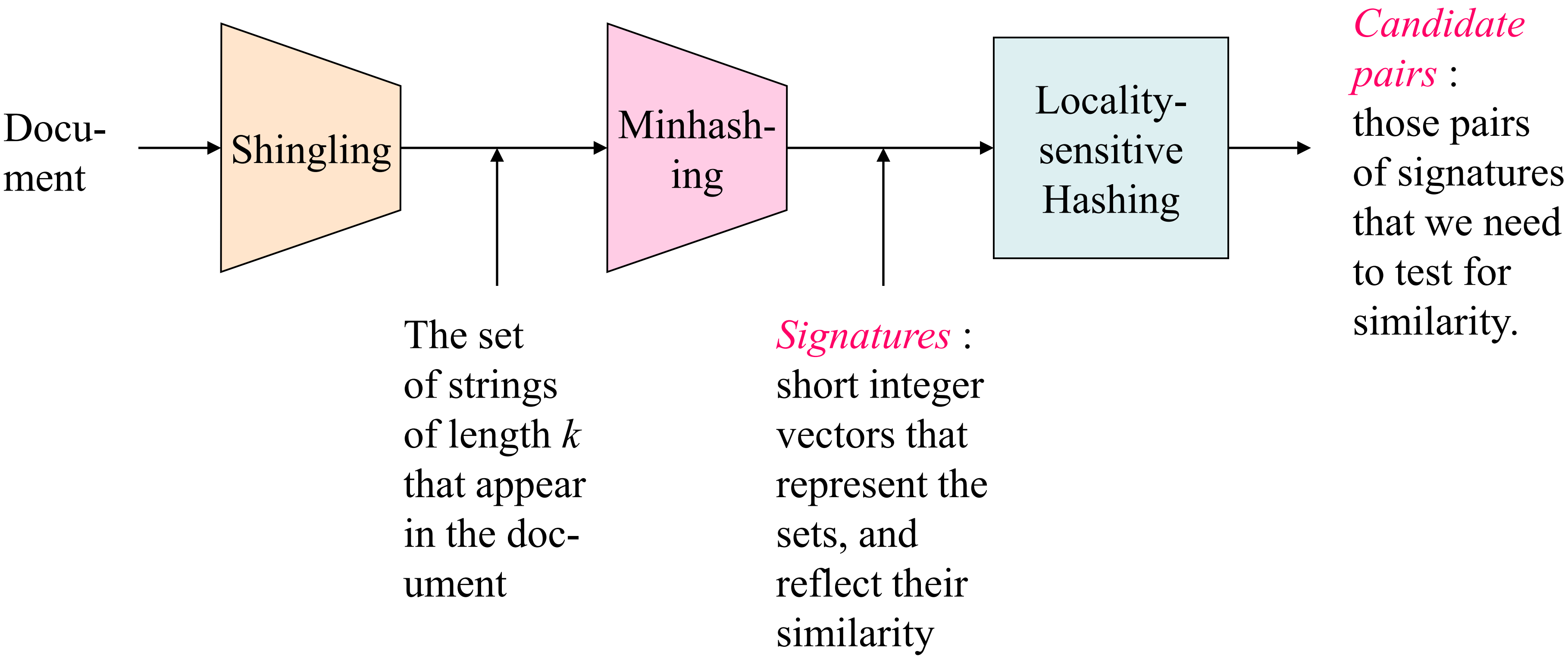
- **Both are *A* /(*A* +*B* +*C*)!**
- **Why?**
  - Look down the permuted columns C_1 and C_2 until we see a 1.
  - If it's a type-*a* row, then *h* (C_1) = *h* (C_2). If a type-*b* or type-*c* row, then not.

Locality-Sensitive Hashing

All-pair comparison is expensive

- We want to compare objects, finding those pairs that are sufficiently similar.
- comparing the signatures of all pairs of objects is quadratic in the number of objects
- Example: 10^6 objects implies $5*10^{11}$ comparisons.
  - At 1 microsecond/comparison: 6 days.

The Big Picture



Document

Shingling

Minhashing

Locality-sensitive Hashing

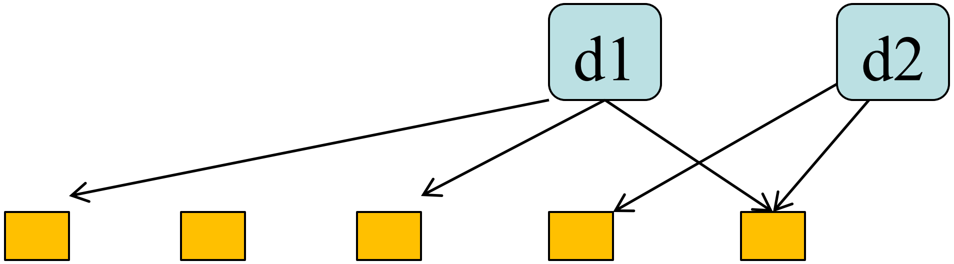
The set of strings of length *k* that appear in the document

Signatures : short integer vectors that represent the sets, and reflect their similarity

Candidate pairs : those pairs of signatures that we need to test for similarity.

Locality-Sensitive Hashing

- **General idea: Use a function f(x,y) that tells whether or not *x* and *y* is a *candidate pair* : a pair of elements whose similarity must be evaluated.**
- **Map a document to many buckets**



- **Make elements of the same bucket candidate pairs.**
  - Sample probability of collision:
    - 10% similarity → 0.1%
    - 1%   similarity → 0.0001%

Application Example of LSH with minhash

Generate b LSH signatures for each url, using r of the min-hash values (${b}$ = 125, r = 3)

- For i = 1...b
  - –Randomly select r min-hash indices and concatenate them to form i'th LSH signature

- **Generate candidate pair (u,v) if u and v have an LSH signature in common in any round**
  - Pr(lsh(u) = lsh(v)) = Pr(mh(u) = mh(v))^r

[Haveliwala, et al.]

Example: LSH with minhash

Document 1:
{mouse, dog, horse, ant}
MH_1 = horse
MH_2 = mouse
MH_3 = ant
MH_4 = dog

LSH_{134} = horse-ant-dog
LSH_{234} = mouse-ant-dog

Document 2:
{cat, ice, shoe, mouse}
MH_1 = cat
MH_2 = mouse
MH_3 = ice
MH_4 = shoe

LSH_{134} = cat-ice-shoe
LSH_{234} = mouse-ice-shoe

Example of LSH mapping in web site clustering

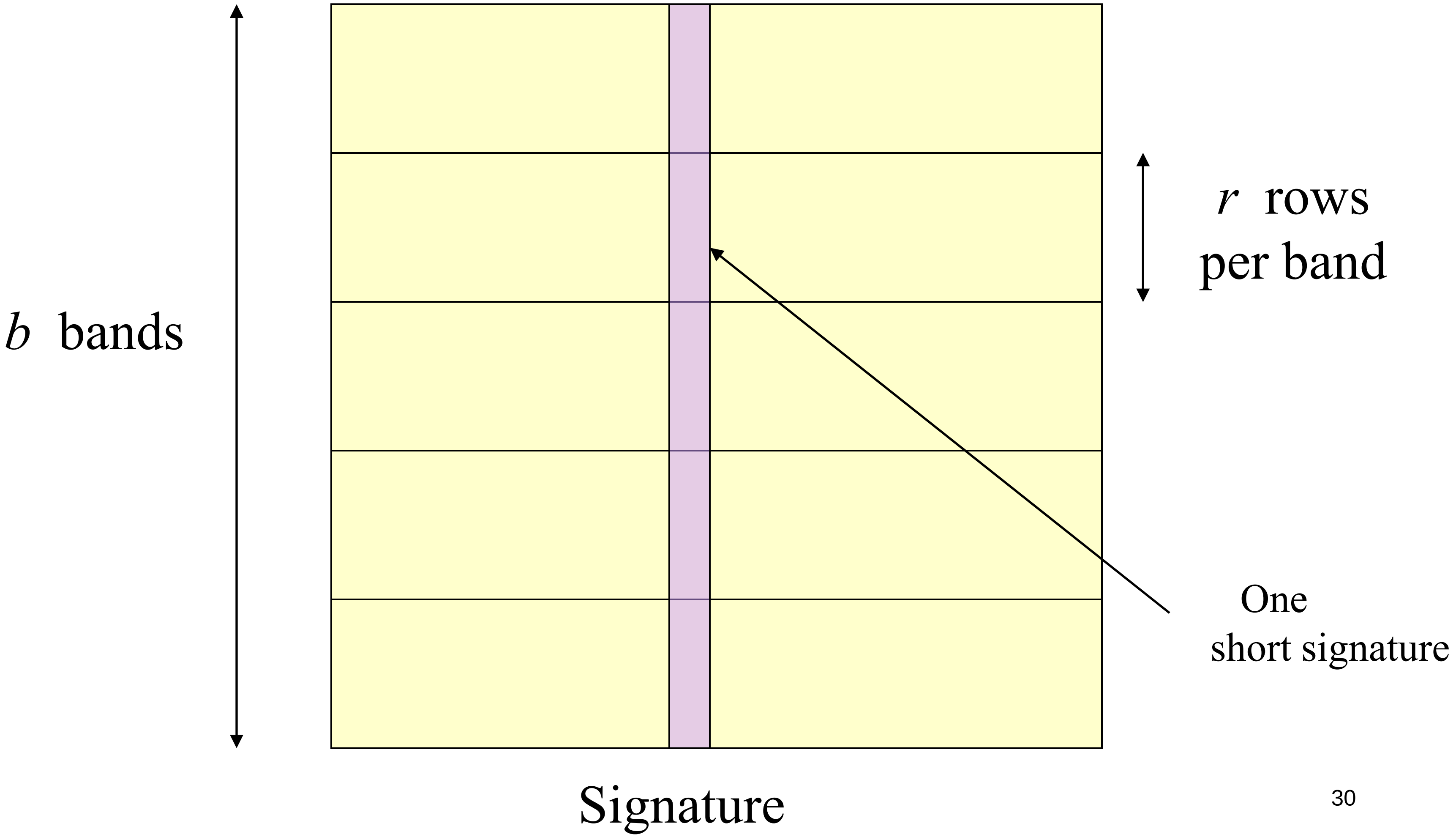
Round 1

sports.com golf.com party.com	. . .	music.com opera.com	. . .	sing.com
sport-team-win		music-sound-play		sing-music-ear

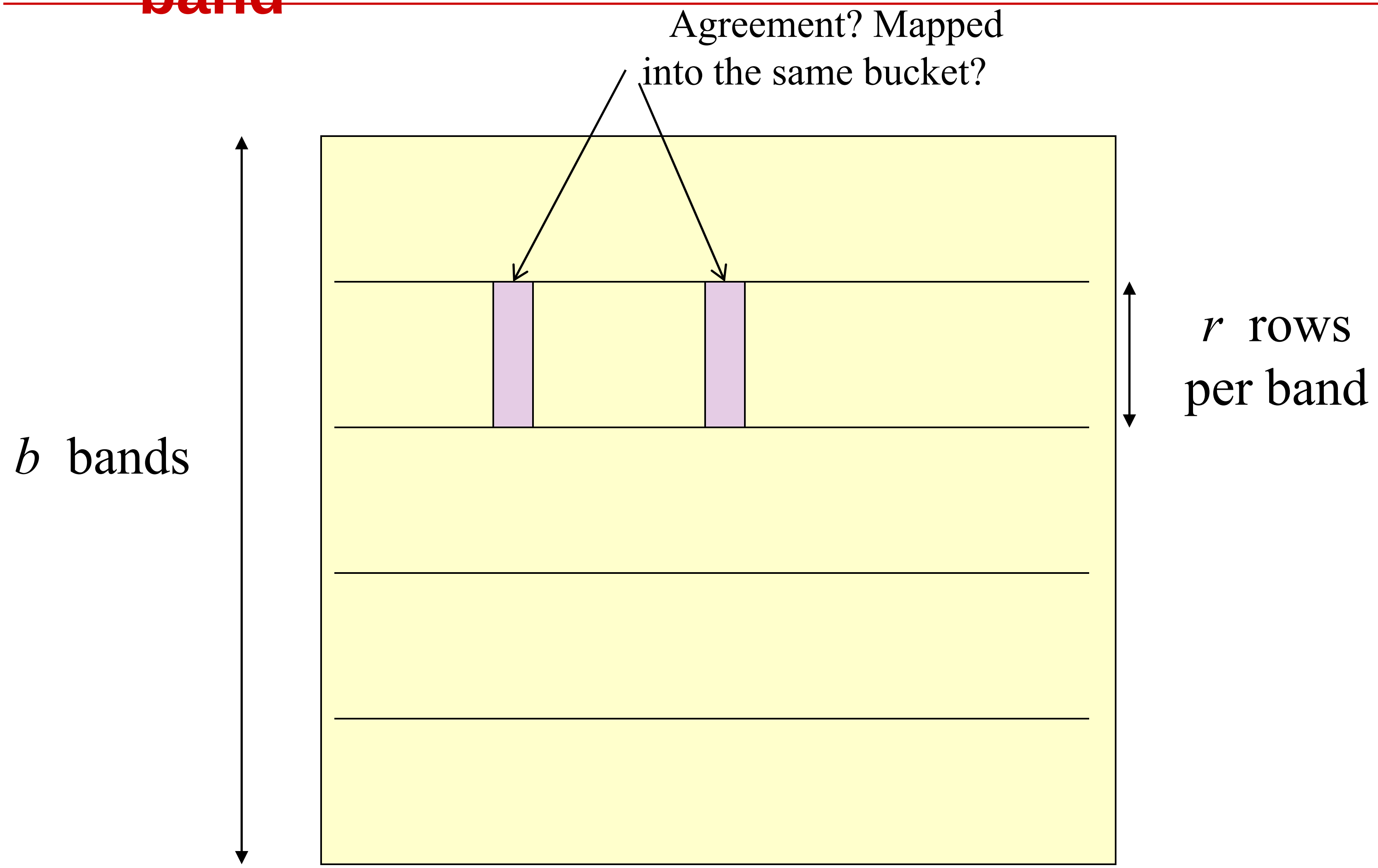
Round 2

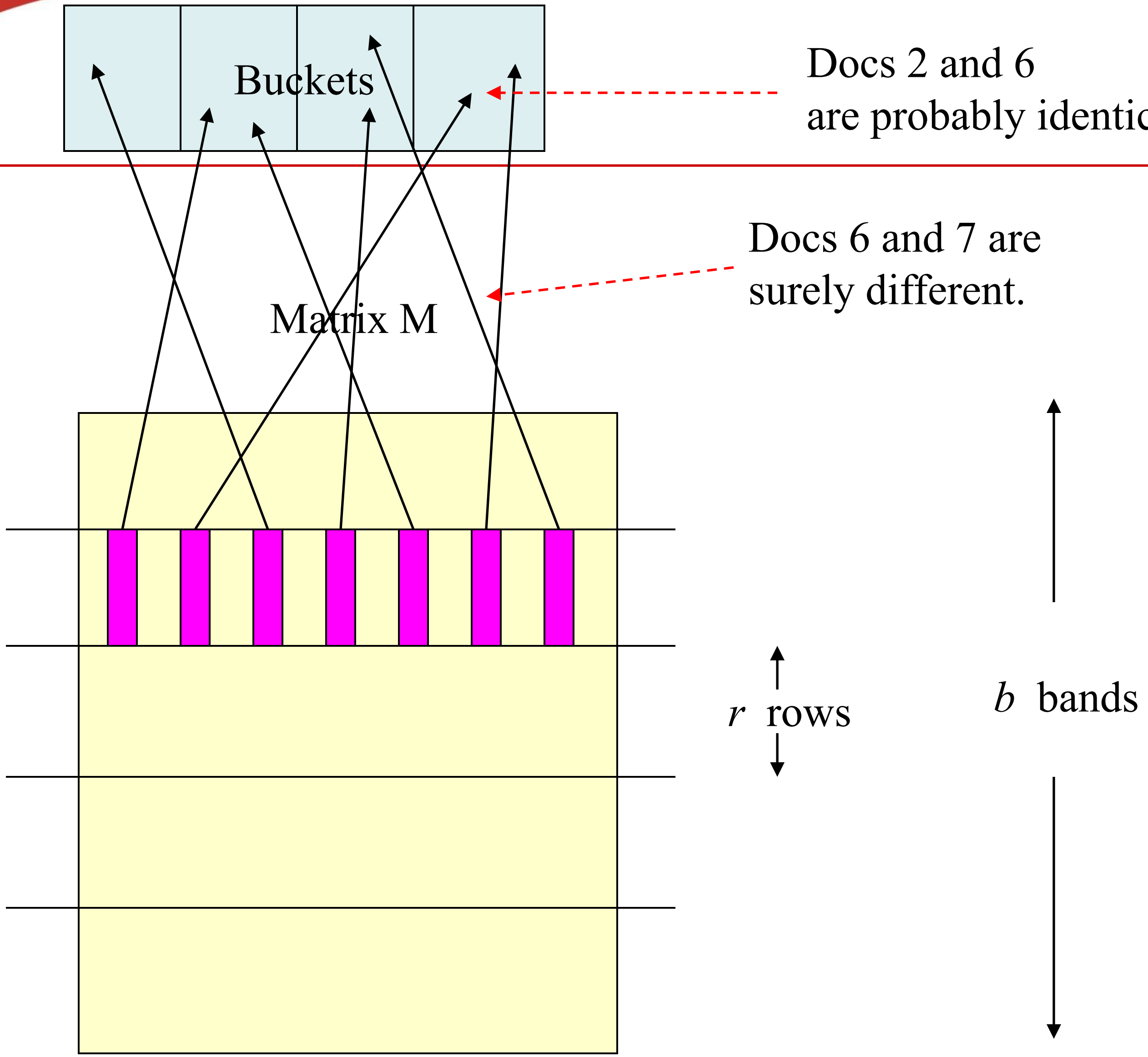
sports.com golf.com	. . .	music.com sing.com	. . .	opera.com
game-team-score		audio-music-note		theater-luciano-sing

Another view of LSH: Produce signature with bands



Signature agreement of each pair at each band





Docs 2 and 6
are probably identical.

Docs 6 and 7 are
surely different.

Signature generation and bucket comparison

- **Create *b* bands for each document**
  - Signature of doc X and Y in the same band agrees → a candidate pair
  - Use *r* minhash values (*r* rows) for each band

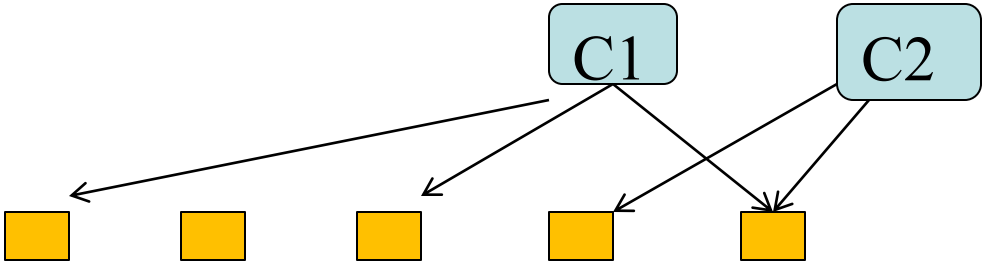
- **Tune *b* and *r* to catch most similar pairs, but few nonsimilar pairs.**

Analysis of LSH

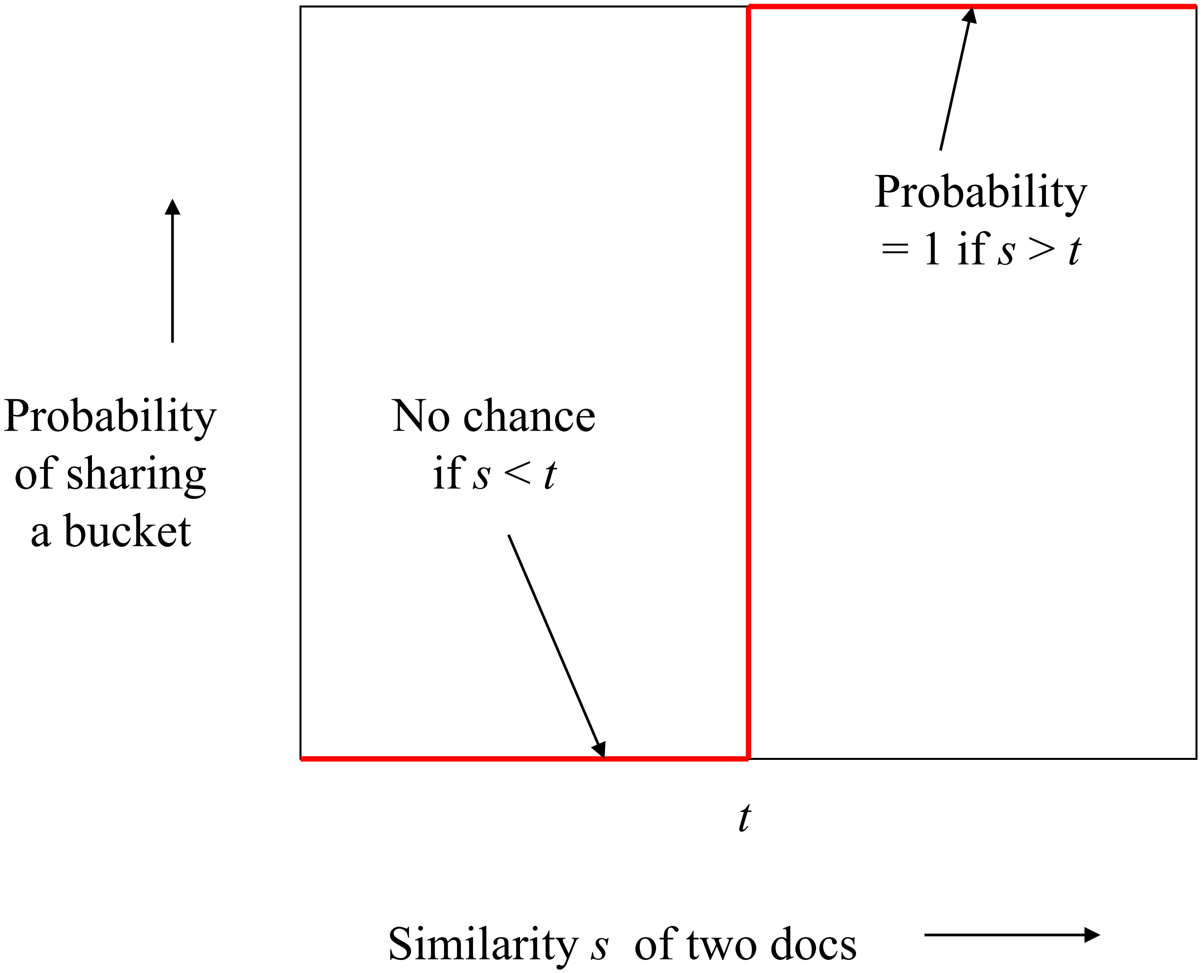
- Probability the minhash signatures of C_1, C_2 agree in one row: s
  - Threshold of two similar documents
- Probability C_1, C_2 identical in one band: s^r
- Probability C_1, C_2 do not agree at least one row of a band: $1-s^r$
- Probability C_1, C_2 do not agree in all bands: $(1-s^r)^b$
  - False negative probability
- Probability C_1, C_2 agree one of these bands: $1-(1-s^r)^b$
  - Probability that we find such a pair.

Example

- Suppose C_1, C_2 are 80% Similar
- Choose 20 bands of 5 integers/band.
- Probability C_1, C_2 identical in one particular band: $(0.8)^5 = 0.328$.
- Probability C_1, C_2 are *not* similar in any of the 20 bands: $(1-0.328)^{20} = .00035$.
  - i.e., about 1/3000th of the 80%-similar column pairs are false negatives.



Analysis of LSH – What We Want



Example: *b* = 20; *r* = 5

Probability of a similar pair to share a bucket

s	$1-(1-s^r)^b$
.2	.006
.3	.047
.4	.186
.5	.470
.6	.802
.7	.975
.8	.9996

LSH Summary

- **Get almost all pairs with similar signatures, but eliminate most pairs that do not have similar signatures.**
  - Check that candidate pairs really do have similar signatures.
- **LSH involves tradeoff**
  - Pick the number of minhashes, the number of bands, and the number of rows per band to balance false positives/negatives.
  - Example: if we had only 15 bands of 5 rows, the number of false positives would go down, but the number of false negatives would go up.