
BRAINLY

About me

hubbry

hubert@brylkowski.com

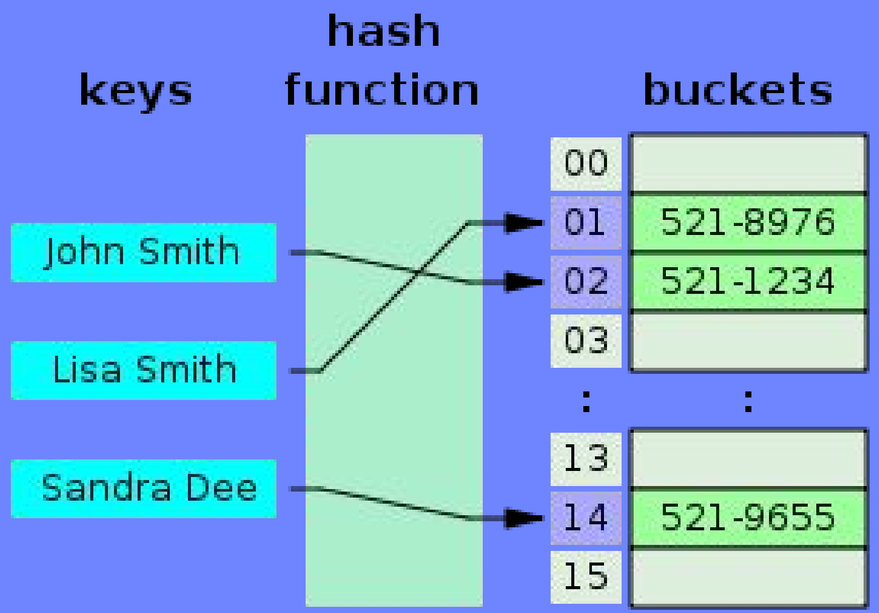
Detecting near-duplicate text messages using LSH and other techniques

Problem - duplicated questions

question_1 == question_2

n^2 comparisons

:(



Use hashmap!

O(n) complexity

What about near duplicates?

- Who was the first king of Poland?
- Who was the first ruler of Poland?

String similarity metrics: Levenshtein distance, Hamming distance, etc. $O(n^2)$ again :(

What if we could get same hashes for similar strings?

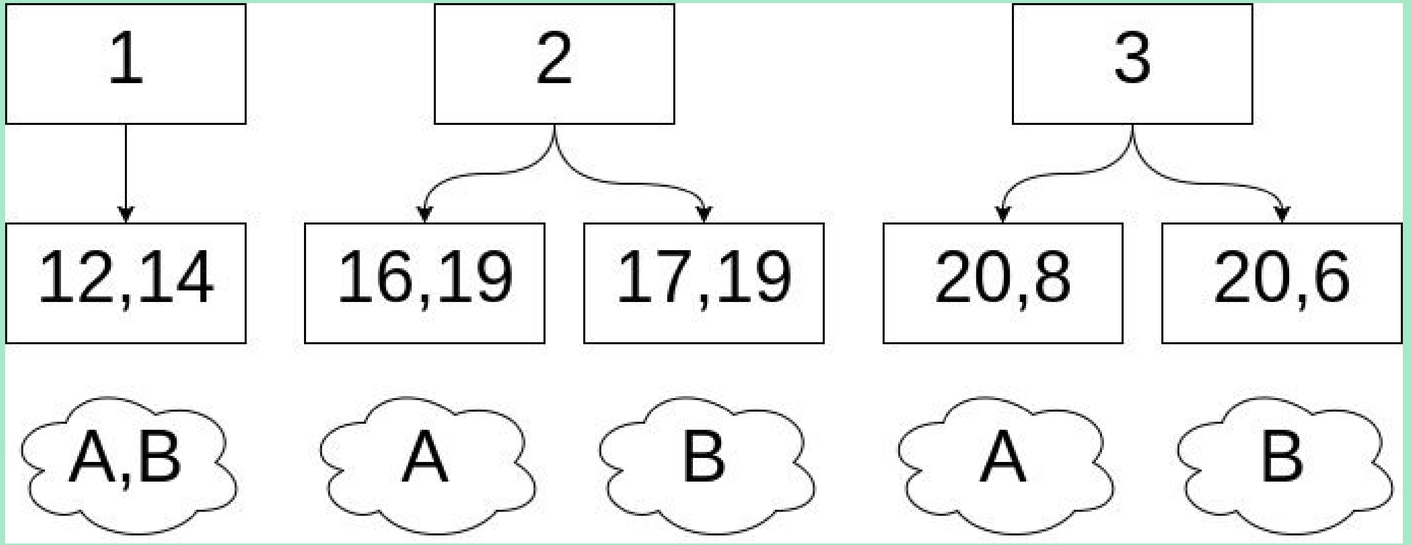
Locality-**S**ensitive **H**ashing

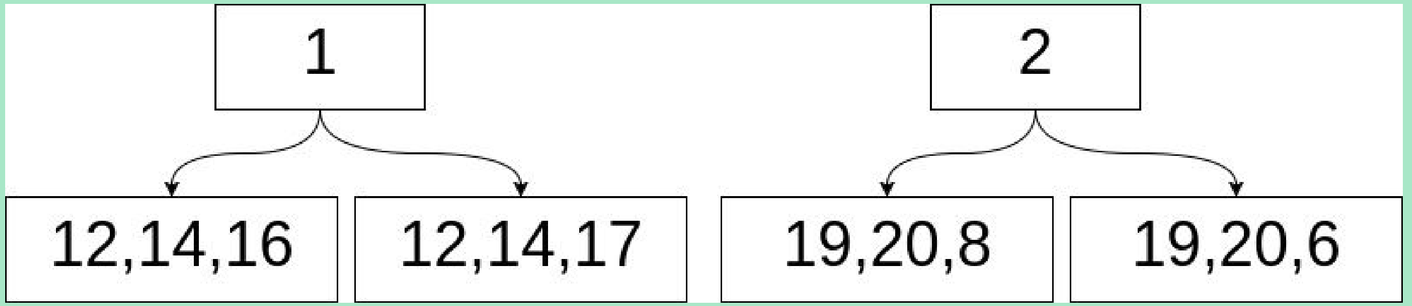
How to do it:

- **For every document:**
  - **Compute minhash**

    [12, 14, **16**, 19, 20, **8**] - **A**

    [12, 14, **17**, 19, 20, **6**] - **B**


1
2
3
12,14
16,19
17,19
20,8
20,6
A,B
A
B
A
B


1
2
12,14,16
12,14,17
19,20,8
19,20,6

In one bin are **POSSIBLE** duplicates

Minhash

Break documents into shingles:

Lorem ipsum

[‘Lore’,’orem’,’rem i’ ...]

Calculate hash for every shingle and find min

Repeat like 200 times with different hash algorithms

Q & A

Further reading:

Lectures about minhash and LSH:

https://www.youtube.com/playlist?list=PL9Al9hamivVmr3GHtfUFkmTkUmicOc9Nn

Python library:

https://github.com/mattilyra/LSH

MinHash for dummies:

http://matthewcasperson.blogspot.com/2013/11/minhash-for-dummies.html