Finding Near-Duplicate Web Pages
A Large-Scale Evaluation of Algorithms

CSCI 599: Content Detection and Analysis for Big Data

Surasit Prakunhungsit

Paper Abstract

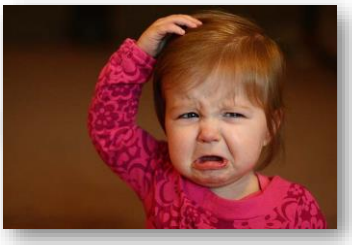
Near-Duplicate Algorithm	Results	Same site Evaluation
• Shingling • Charikar's	• Both Good for "Different sites" • Neither works on "Same sites"	• 50% precision for Charikar's • 38% for Shingling • 79% for a combined algorithm

USC Viterbi School of Engineering
University of Southern California

Problems

Waste space

Slow Down



Annoy the users

Related Works

Native Solution	Broder et al. (Shingling)	Charikar
• Compare all pairs to document • Expensive on large datasets	• Use word sequences to efficiently find near-duplicate pages	• Random projections of words in the document

Why should we care?

Moss (Measure Of Software Similarity)

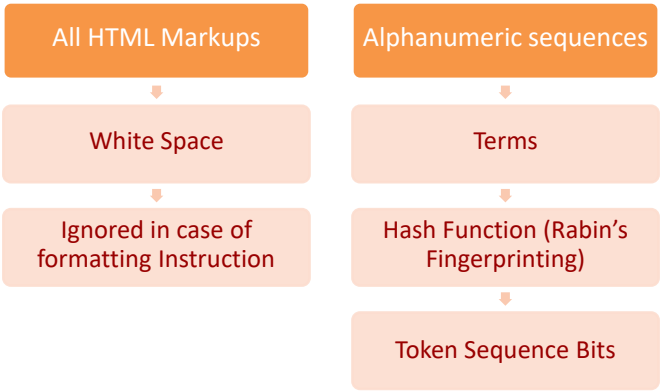
Search Engines

Email Spam Detection

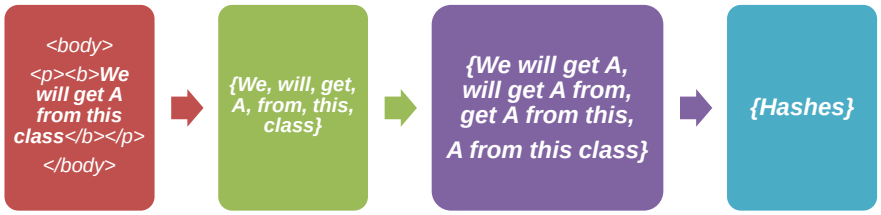
Relation to the Class Lecture

Shingling and Charikar Similarities

- Creation of ***token sequences***



Shingling Example



Web page document

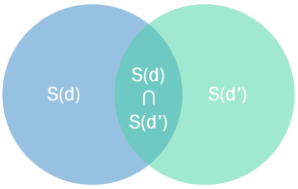
Length ***n***
Subsequence ***k*** tokens
n = 7
k = 4

Results in 4-shingles: sequence of
n – k + 1 fingerprints

{I, j, k, l}

Shingling

- The percentage of unique shingles on which the two pages agree is a good measure for the similarity

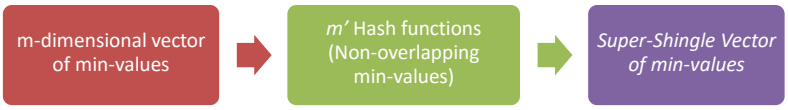


$$\frac{|S(d) \cap S(d')|}{|S(d) \cup S(d')|}$$

- The percentage of entries in the min-values vector that two pages agree could be use for a similarity approximation

Shingling

- To save space and speed up



{l, j, k, x, y, z}
Each 64 bits
48 Bytes per page

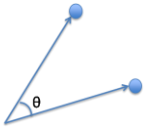
- ***B-similarity*** is the number of identical entries in the **super-shingle vectors** from two pages
  - Two pages are near-duplicates iff their B-similarity is at least 2

Page d = {l, j, k, x, y, z}
Page d' = {a, b, c, x, y, p}

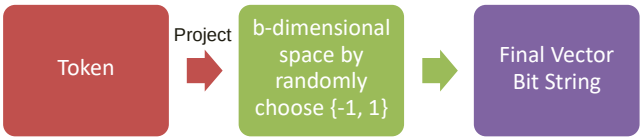
Charikar

- The ***cosine similarity*** of two pages is a good measure for the similarity

$$sim(A,B) = \cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|}$$



- The ***cosine similarity*** of two pages is ***proportional*** to the ***number of bits*** in which the two **projections** agree.



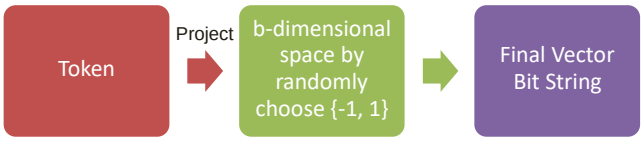
b = 384 bits
48 Bytes per page

$$h_{\vec{r}}(\vec{u}) = \begin{cases} 1 & \text{if } \vec{r} \cdot \vec{u} \geq 0 \\ 0 & \text{if } \vec{r} \cdot \vec{u} < 0 \end{cases}$$

Result = 100111001….

Charikar

- ***C-similarity*** of two pages is the number of bits their projections agree on
- Two page are near-duplicate iff the number of agreeing bits in their projections goes above a fixed ***threshold t (t = 372)***



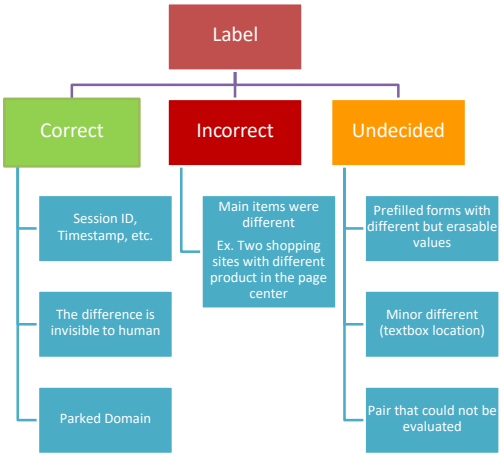
b = 384 bits
48 Bytes per page

$$h_{\vec{r}}(\vec{u}) = \begin{cases} 1 & \text{if } \vec{r} \cdot \vec{u} \geq 0 \\ 0 & \text{if } \vec{r} \cdot \vec{u} < 0 \end{cases}$$

Result1 = 100101001….
Result2 = 100111000….

Human Evaluation

- Randomly sample B-similar and C-similar pairs



Result for Shingling

Near dups	Number of pairs	Correct	Not correct	Undecided
all	1910	0.38	0.53	0.09 (0.04)
same site	1758	0.34	0.57	0.09 (0.04)
diff. sites	152	0.86	0.06	0.08 (0.01)
B-sim 2	1032	0.24	0.68	0.08 (0.03)
B-sim 3	389	0.42	0.48	0.1 (0.04)
B-sim 4	240	0.55	0.36	0.09 (0.05)
B-sim 5	143	0.71	0.23	0.06 (0.02)
B-sim 6	106	0.85	0.05	0.1 (0.08)

Table 1: Shingling Fraction of correct, not correct, and undecided pairs

Result for Charikar

Near dups	Number of pairs	Correct	Not correct	Undecided
all	1872	0.50	0.27	0.23 (0.18)
same site	1393	0.36	0.34	0.30 (0.25)
different site	479	0.90	0.05	0.05 (0)
C-sim $\geq$ 382	179	0.47	0.37	0.16 (0.10)
382 > C-sim $\geq$ 379	407	0.40	0.37	0.23 (0.18)
379 > C-sim $\geq$ 376	532	0.37	0.27	0.35 (0.30)
C-sim $<$ 376	754	0.62	0.19	0.19 (0.12)

Table 2: Charikar Fraction of correct, not correct, and undecided pairs

Result Comparison: Shingling and Charikar

Overall

- Charikar outperform Shingling (50% > 38%)

Pairs on different sites

- Both achieve high precision
- Charikar is superior to shingling (90% > 86%)

Pairs on the same site

- Neither achieve high precision
- Charikar achieves slightly higher precision (36% > 34%)

USC Viterbi School of Engineering
University of Southern California

The Combined Algorithm

- First compute all B-similar pairs.
- Then filter out those pairs whose C-similarity falls below a certain threshold.

Near dups	Number of pairs	Correct	In-correct	Un-deci-ded
all	363	0.79	0.15	0.06
same site	296	0.74	0.19	0.07
different site	65	0.99	0.00	0.01

Table 3: Combined Algorithm Fraction of correct, not correct, and undecided pairs

Conclusion

Combined Algorithm outperform both Charikar and Shingling

(79% > 50% > 38%)

Parameters ***k*** and ***t*** affect the result precision

Charikar & Shingling perform poorly for the same site due to ***boilerplate text***.

We could use a **boilerplate detection algorithm** to resolve this issue

Pros and Cons

Pros

- Clear tables and beautiful graphs
- Good organization

Cons

- No examples of the two main algorithms
- Wrong spelling

larity computation the m-dimensional vector of minvalues is reduced to a m'-dimensional vector of *supershingles* by fingerprinting non-overlapping sequences of minvalues: Let m be divisible by m' and let $l = m/m'$. The concatentation of minvalue $j*l, \ldots, (j+1)*l-1$ for $0 \leq j < m'$ is fingerprinted with yet another fingerprinting function and is called *supershingle*.[2] This creates a supershingle vector. The number of