AGENDA

• Papers
  • Overview
  • Methods
  • Experiments

• Discussion
IDENTIFYING AND FILTERING NEAR-DUPLICATE DOCUMENTS

(Broder 2000)
Overview

- Introduces a method for representing a document as a fixed size “sketch”

- Sketches can be used to quickly estimate document-document “resemblance”
Method

- Break a document into shingles
- Fingerprint each shingle
- Take the min for t random permutations
- Estimate resemblance between two documents using the min values
Results

- Broder’s technique can rid a collection of its near-duplicates in $O(m \log m)$ time
- Has been used by AltaVista
Finding near-duplicate web pages: a large-scale evaluation of algorithms

(Henzinger 2006)
Two document similarity algorithms (Broder’s and Charikar’s) are applied to 1.6B web pages and measured for performance

- Broder’s .38 vs. Charikar’s .5 (precision)
- The algorithms performed poorly on pages from the same domain
- A combined algorithm is presented where the precision is .79 (recall .79 of others)
Methods

- Super shingles are shingles of shingles
- They save space and time
- Similarity is $|A \cap B|$
- Parameters: $m, m', l, k$
Methods

- Charikar’s Random Projections
- Project feature vectors onto a random vector
- Set negative values to 0, positive to 1
- Number of agreeing bits is proportional to the cosine similarity between two documents
Experiments

- Used Mapreduce to distribute the computation
- Preprocessing removed all but one of each type of exact duplicate
- 25%-30% of the documents were removed
Experiments

- Of the 1.6B pages
- 42.6M hosts
- Average of 36.5 pages/host
- pages/host follow a power law
Experiments

- Alg B and C were tuned so that they would return roughly the same number of docs.

- Due to super shingles, B is more likely to return documents with more consecutive term differences.

- 28% of B’s incorrect pairs were caused by pages from two databases, why?
Experiments

- The combined algorithm computed B-similar pairs and then used alg C
- Made up for B’s biggest flaw
- Percentage of correct near-duplicates found: ~%80
- Precision of the combined algorithm: ~.77
Exploiting sentence-level features for near-duplicate document detection

(Wang & Chang 2009)
These authors attempt to identify documents which are “mutually inclusive”

Easier

They use two sentence-level techniques to determine similarity
Method

Sentence-level fingerprinting

- Instead of n-grams, they hash each sentence

Sentence length

- n-grams with doc as sentence lengths

Dynamic jumping
Note:

“The feature string is a sequence of sentence fingerprints, and each feature vector is simply one hashed sentence”

Very confusing
When performing their experiments they compared a few other methods:

- Full Fingerprinting -- `sha1` on 6-grams
- Shingling -- hashed one in every 3/6 gram
Experiments

- Used six corpora from the CLIR task in NTCIR workshop (English)

- Total documents: 182,067

- Randomly selected 180 documents as the query set

- Appended sentences from each query doc to the same document
Experiments

Table 3 shows computational performance

- Sentence-level procedures were fastest
- They took up less memory
- They generated fewer features
- Other algorithms were not “optimal”
Experiments

- Fig. 4 shows precision and F1 values for the top-k ranked results
- Table 5 shows identification performance
- Almost everything performed well
Conclusion

For some tasks, sentence-level features can give comparable results while conserving resources.

When would it be good to use sentence-level features?
A short survey of document structure similarity algorithms

(Buttler 2004)
Instead of using content, these authors use web structure to approximate similarity.
Methods

- Tree edit distance
- Weighted tag similarity
- Fourier transform
- Path shingles and similarity
Experiments

- Clustering 500 synthetic XML documents
- They approximate TED
- Error is seen as a classification that differs from TED’s
- Path similarity performs best
Experiments

- Clustering web pages from various sites
- Clustered based on domain
- Weighted tag similarity performs best
- TED (their metric) did not perform well
Experiments

- Clustering pages from my.yahoo.com
- FFT performs most similar to TED with 6 clusters
- Other methods perform better with 2 clusters
The approximation algorithms are much faster and they have comparable accuracy.
Winnowing: Local Algorithms for Document Fingerprinting

(Schleimer, Wilkerson, & Aiken 2003)
Figure 3.8: The winnowing chunk selection heuristic. All chunks are hashed; a fixed-length window is then passed over the hash values and all chunks whose hash is minimal for some window in which it appears are marked. All marked chunks are selected for the fingerprint.
Sizing sketches: a rank-based analysis for similarity search

Wang, Dong, ... (2007)
Authors suggest using sketches for comparing data types of high dimension

Use them as a filtering mechanism for images, audio, 3D shapes