New Issues in Near-duplicate Detection

Martin Potthast and Benno Stein

Bauhaus University Weimar
Web Technology and Information Systems

Motivation

About 30% of the Web is redundant. [Fetterly 03, Broder 06]

Content redundancy occurs in various forms:

- Mirrors.
- Crawl artifacts, such as the same text with a different date or a different advertisement, available through multiple URLs.
- Versions created for different delivery mechanisms (HTML, PDF, etc.)
- Annotated and unannotated copies of the same document
- Policies and procedures for the same purpose in different legislatures
- “Boilerplate” text such as license agreements or disclaimers
- Shared context such as summaries of other material or lists of links
- Syndicated news articles delivered in different venues
- Revisions and versions
- Reuse and republication of text (legitimate and otherwise)

[Zobel 06]

Motivation

About 30% of the Web is redundant. [Fetterly 03, Broder 06]

Content redundancy occurs in various forms:

- Mirrors.
- Crawl artifacts, such as the same text with a different date or a different advertisement, available through multiple URLs.
- Versions created for different delivery mechanisms (HTML, PDF, etc.)
- Annotated and unannotated copies of the same document
- Policies and procedures for the same purpose in different legislatures
- “Boilerplate” text such as license agreements or disclaimers
- Shared context such as summaries of other material or lists of links
- Syndicated news articles delivered in different venues
- Revisions and versions
- Reuse and republication of text (legitimate and otherwise)

[Zobel 06]

Nearly exact copies and modified copies with high similarity.

➜ Near-duplicate documents.

Motivation

Contributions of near-duplicate detection to real-world tasks:

- Index size reduction
- Search result cleaning
- Web crawl prioritization
- Plagiarism analysis

Our contributions to near-duplicate detection:

- Classification of near-duplicate detection algorithms
- Presentation of a new tailored corpus for evaluation
- Comparison of current algorithms
  (including so far unconsidered hashing technologies)

Formalization

Consider a set of documents D. Given a document d_q:

Find all documents $D_q \subset D$ with a high similarity to d_q.

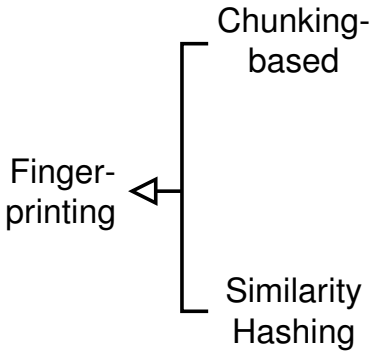
➜ Naive approach: Compare d_q with each $d \in D$.

In detail:
Construct document models for D and d_q, obtaining $\mathbf{D}$ and $\mathbf{d_q}$.
Employ a similarity function $\varphi : \mathbf{D} \times \mathbf{D} \rightarrow [0, 1]$.

- Near-duplicate detection algorithms rely on purposefully constructed document models, called *fingerprints*.
- A fingerprints is a set of k natural numbers, which are computed on the basis document extracts.
- Two documents are considered as duplicates if their fingerprints share at least k_d, $k_d < k$, numbers.

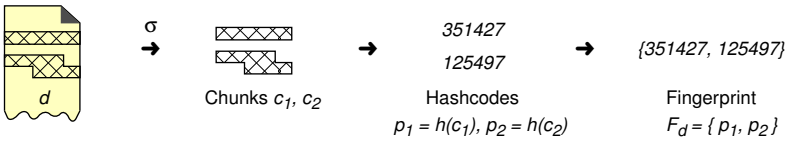
Taxonomy of Fingerprinting Algorithms

Fingerprinting methods



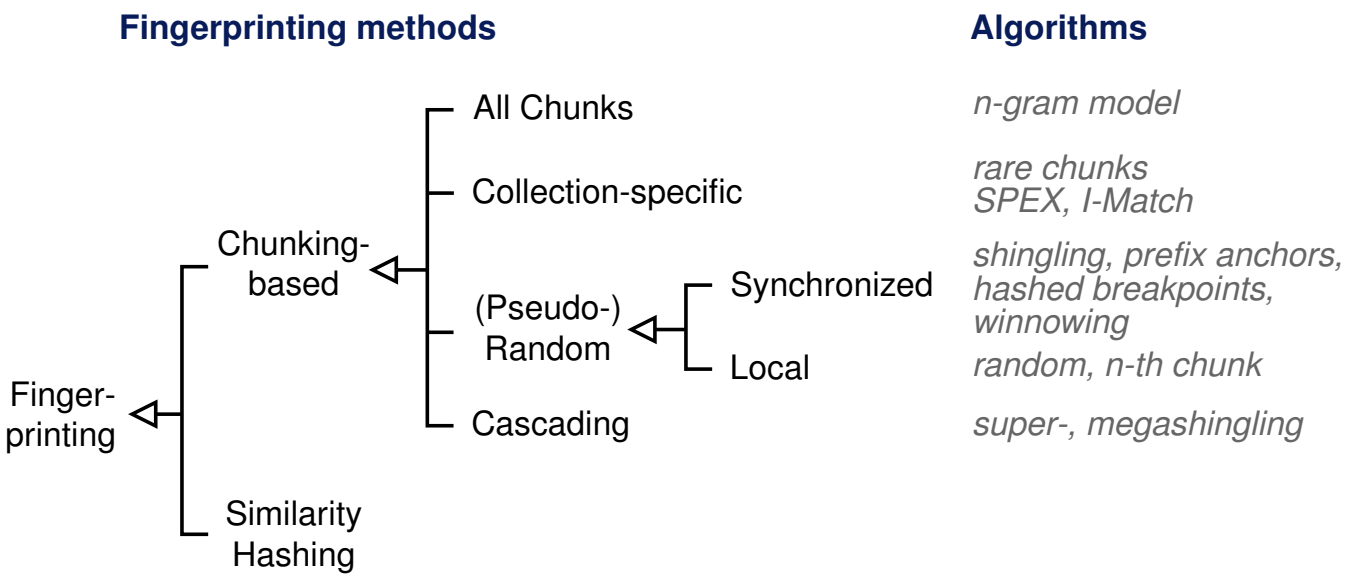
Chunking:

k chunks are selected from a document $\mathbf{d}$.



Chunks are also called n-grams or shingles.

Taxonomy of Fingerprinting Algorithms



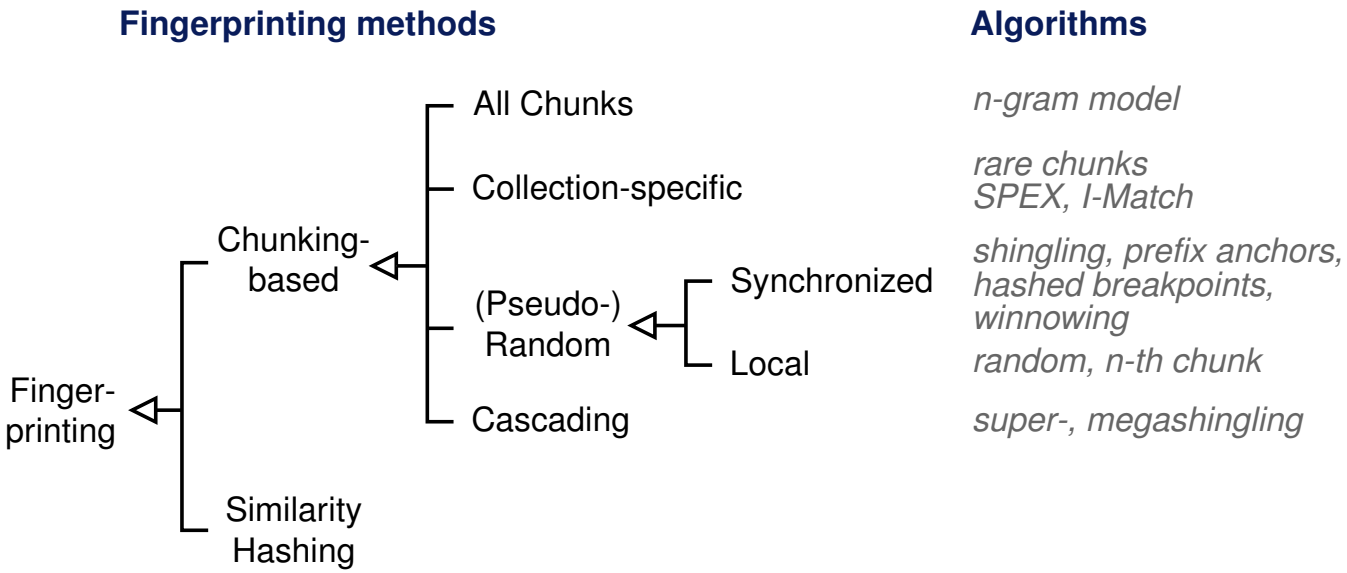
Chunking:

k chunks are selected from a document $\mathbf{d}$.

Selection heuristics:

- all
- based on knowledge about D
- intelligent random choices

Taxonomy of Fingerprinting Algorithms



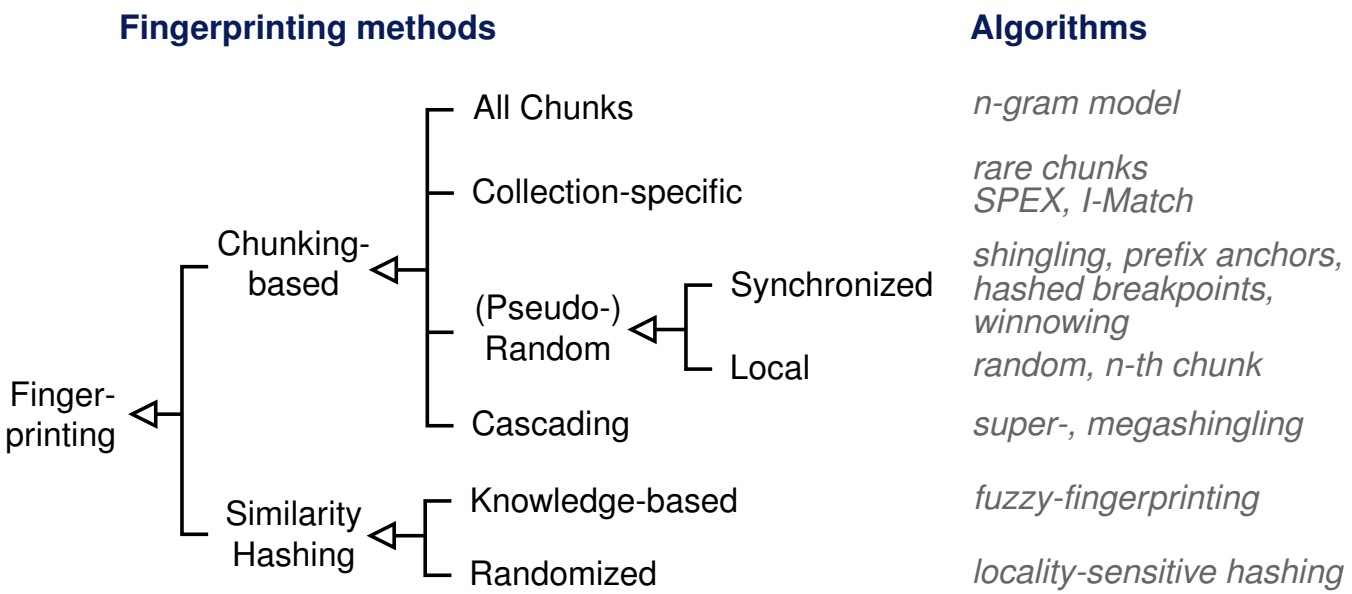
Similarity Hashing:

k particular hash functions $h_\varphi : \mathbf{D} \rightarrow U$, $U \subset \mathbf{N}$, with the property

$$h_\varphi(\mathbf{d}) = h_\varphi(\mathbf{d_q}) \;\Rightarrow\; \varphi(\mathbf{d}, \mathbf{d_q}) \geq 1 - \varepsilon \qquad \text{with } \mathbf{d} \in \mathbf{D}, 0 < \varepsilon \ll 1$$

are used to generate k hashcodes for a document $\mathbf{d}$.

Taxonomy of Fingerprinting Algorithms



Similarity Hashing:

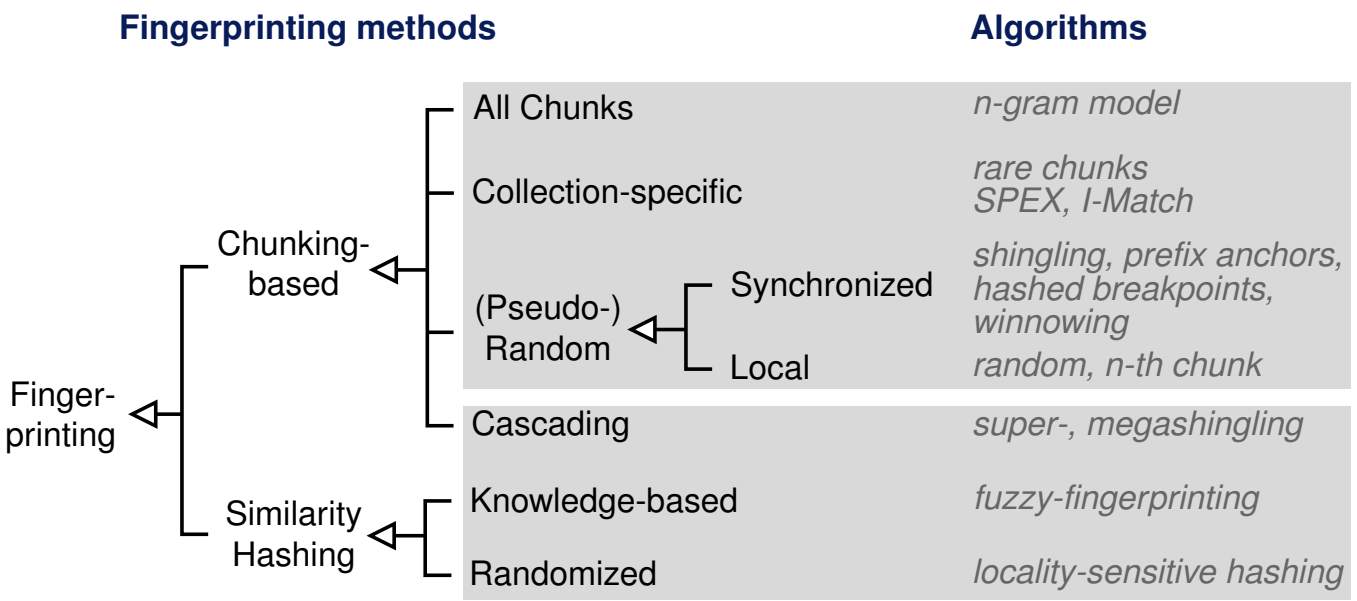
k particular hash functions $h_\varphi : \mathbf{D} \rightarrow U$, $U \subset \mathbf{N}$, with the property

$$h_\varphi(\mathbf{d}) = h_\varphi(\mathbf{d_q}) \;\Rightarrow\; \varphi(\mathbf{d}, \mathbf{d_q}) \geq 1 - \varepsilon \qquad \text{with } \mathbf{d} \in \mathbf{D}, 0 < \varepsilon \ll 1$$

are used to generate k hashcodes.

Hash function construction: domain knowledge vs. randomization.

Taxonomy of Fingerprinting Algorithms



For algorithms in the upper box fingerprints have to share more than one number, $k_d > 1$, to be recognized as duplicates

For algorithms in the lower box fingerprints need to share only one number, $k_d = 1$, to be recognized as duplicates.

(Cascaded) Chunking

(Super-)Shingling (SSh) [Broder 97]

$$\begin{pmatrix} w_1 & w_2 & w_3 \\ w_2 & w_3 & w_4 \\ w_3 & w_4 & w_5 \\ \vdots & & \\ w_{m-2} & w_{m-1} & w_m \end{pmatrix}$$

n-gram vector space model

$$\begin{pmatrix} 12354 \\ 15695 \\ 59634 \\ \vdots \\ 43586 \end{pmatrix}$$

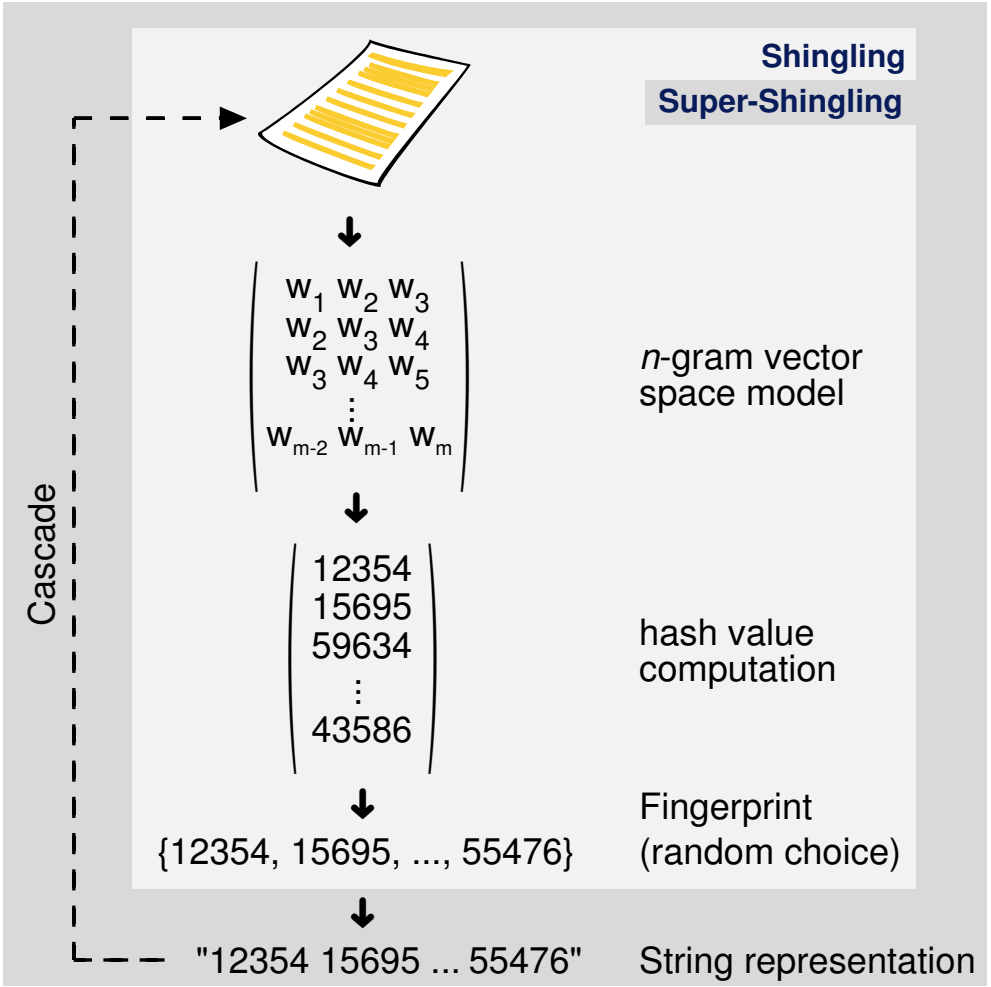
hash value computation

{12354, 15695, ..., 55476}

Fingerprint (random choice)

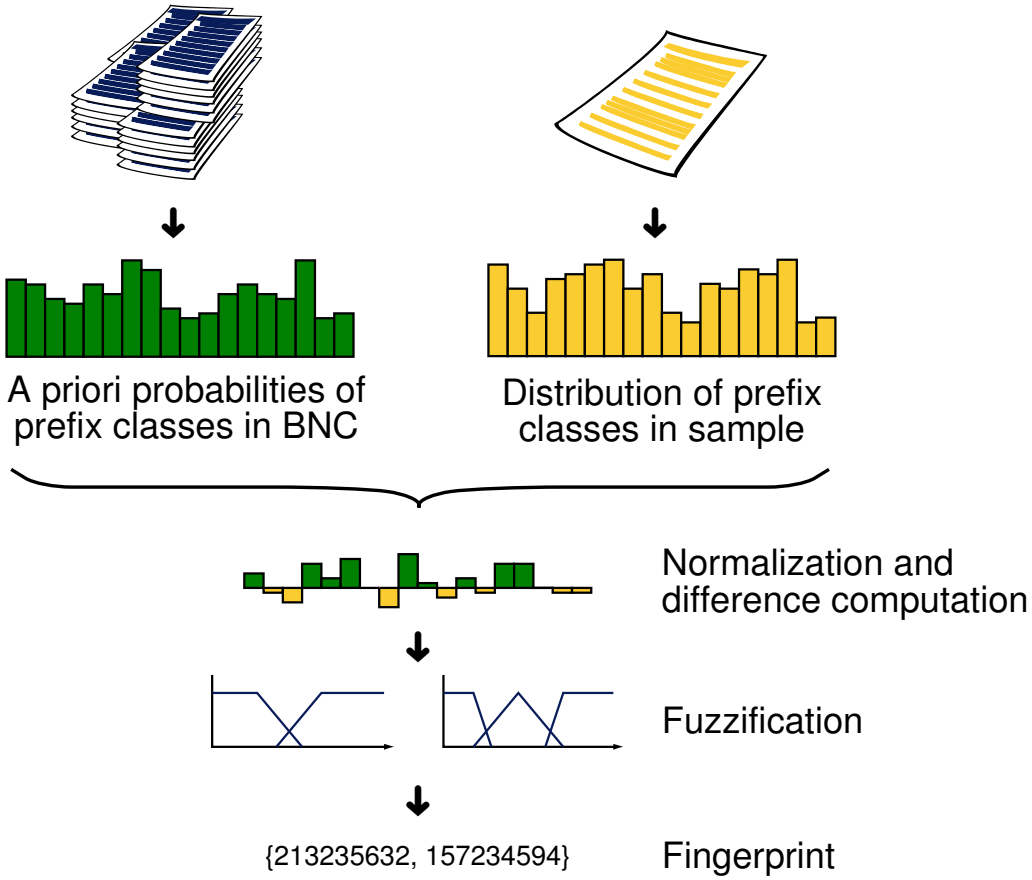
(Cascaded) Chunking

(Super-)Shingling (SSh) [Broder 97]



Similarity Hashing

Fuzzy-Fingerprinting (FF) [Stein 05]

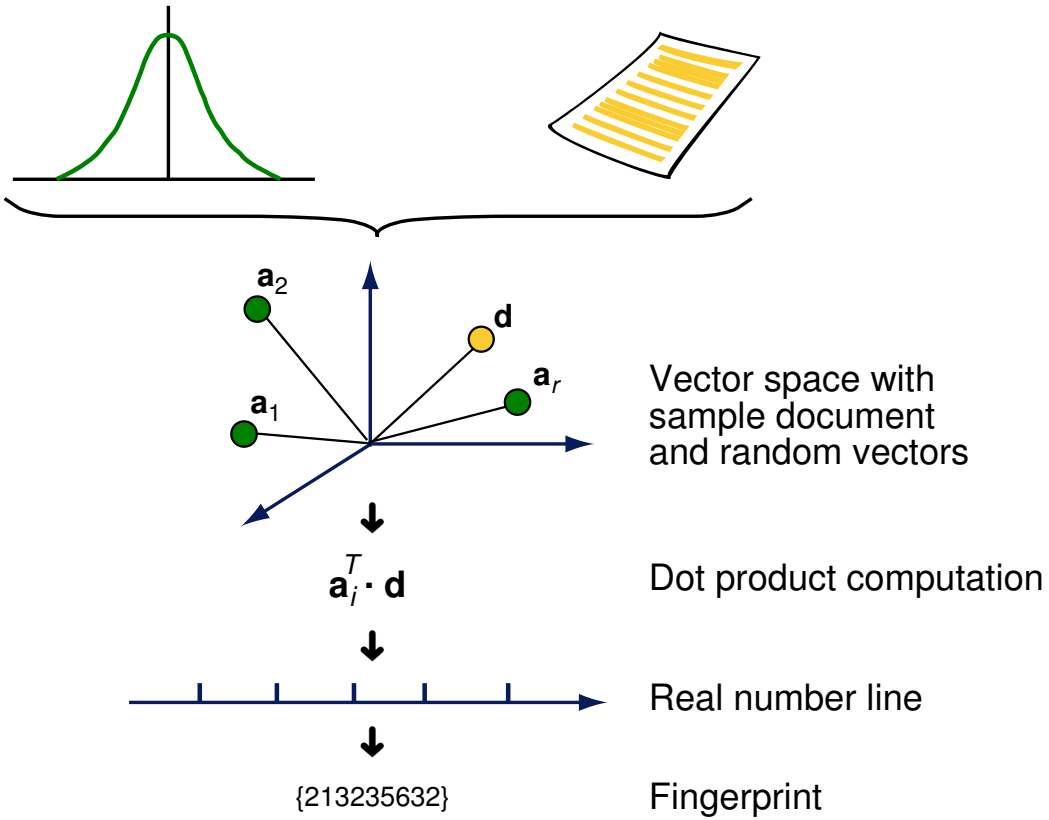


All words having the same prefix belong to the same prefix class.

Similarity Hashing

Locality-Sensitive Hashing (LSH)

[Indyk and Motwani 98, Datar et. al. 04]



The results of the r dot products are summed.

Evaluation Corpus

Wikipedia Snapshot including all Revisions

Existing standard corpora (TREC, Reuters) are not suited for large-scale near-duplicate detection algorithm evaluations.

Wikipedia is a rich resource of versioned and revisioned documents.

Benchmark data:

- approx. 6 million pages (documents)
- approx. 80 million revisions
- XML file of approx. 1 TB

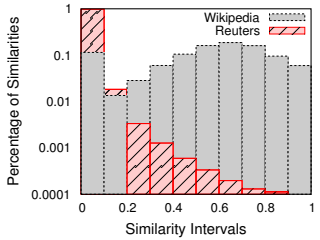
Evaluation Corpus

Wikipedia Snapshot including all Revisions

Experiments:

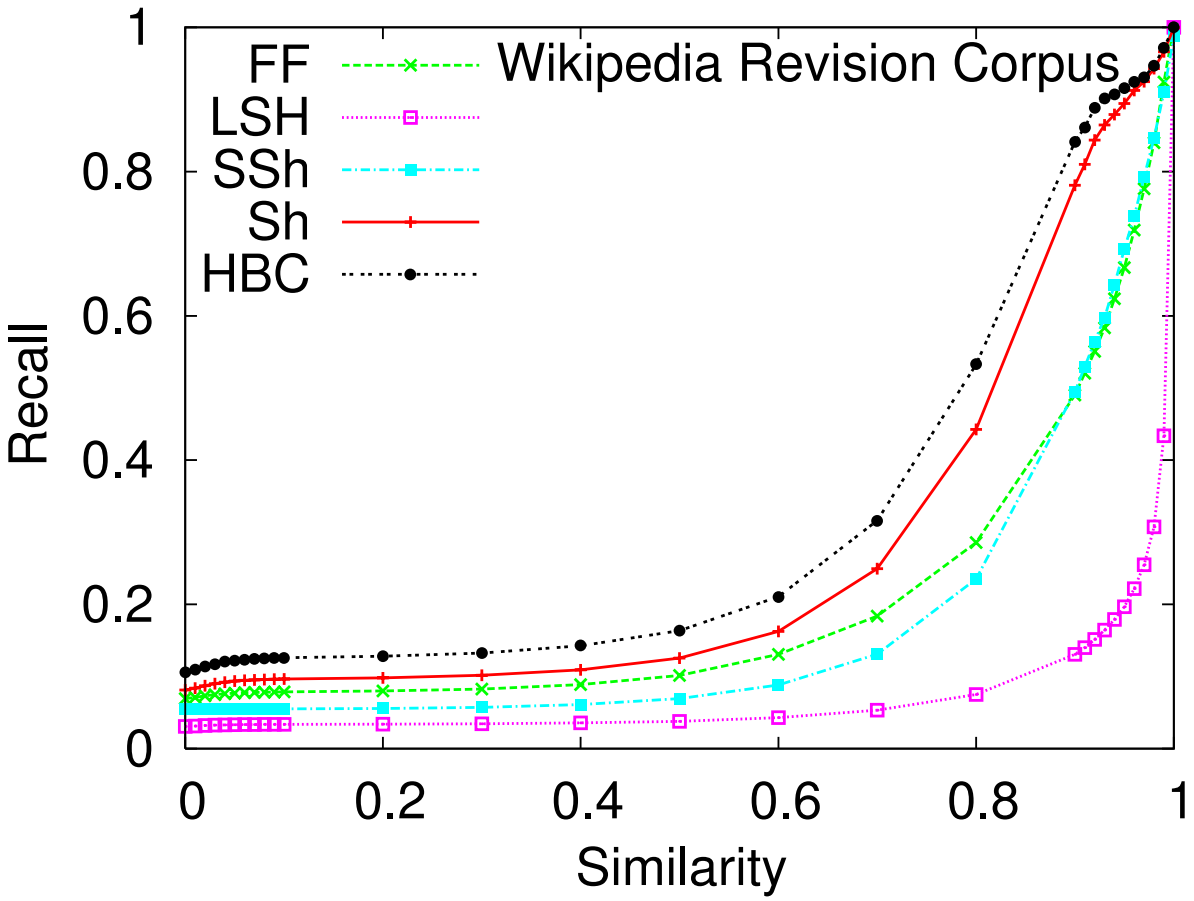
- first revision of each Wikipedia page is in the role of d_q
- d_q was compared with each of it's revisions
- d_q was compared with it's immediate succeeding page

Reference:
Vector space model
with tf and cos-similarity.

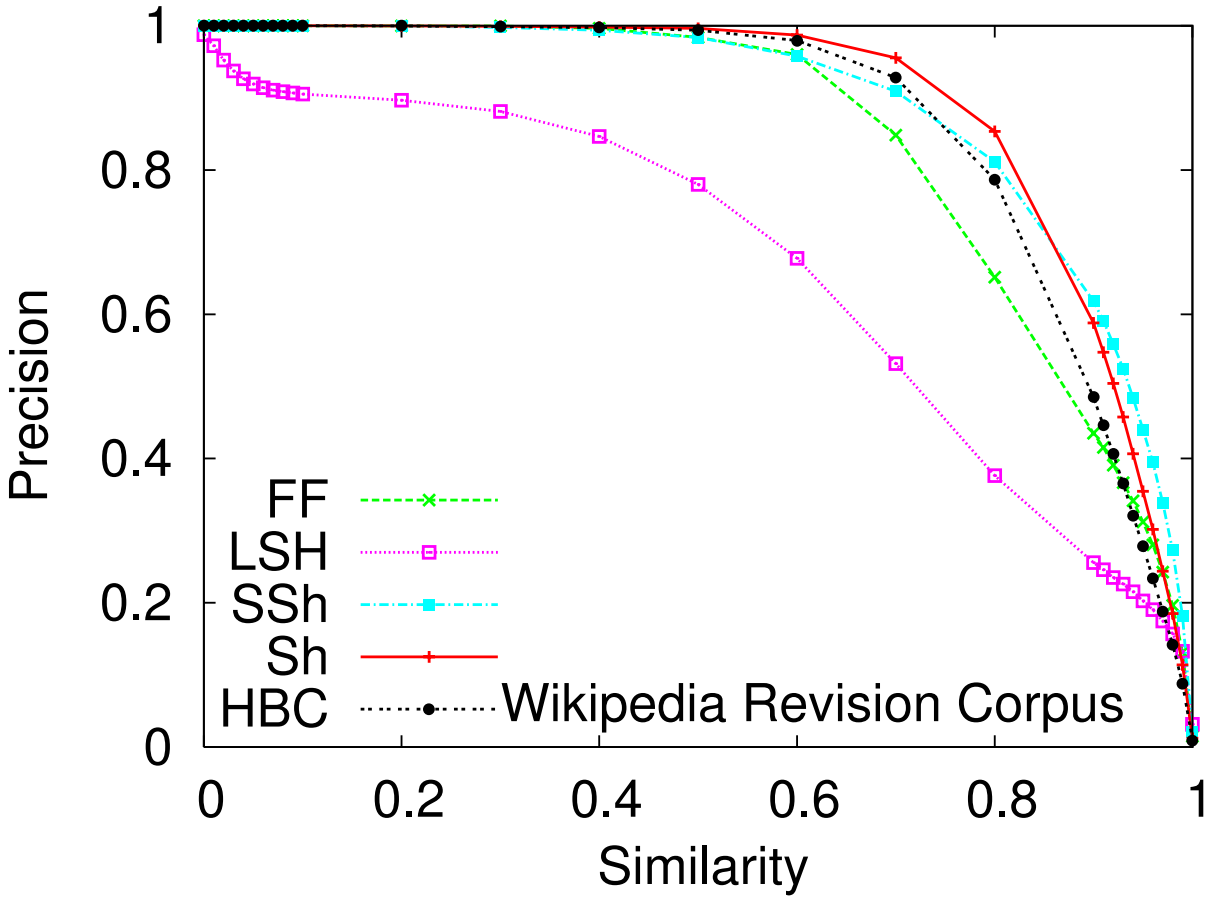


Precision and recall were recorded for similarity thresholds ranging from 0 to 1.

Evaluation Results



Evaluation Results



Summary

Near-duplicate detection accuracy:

- FF outperforms other algorithms in terms of recall.
- No algorithm outperforms another in terms of precision.
- LSH performs poor in both cases.

Wikipedia Revision Collection:

- May be a new standard for high similarity evaluations.
- Allows for evaluations at the Web scale.

Conclusions:

- ➜ Similarity hashing is a promising technology for near-duplicate detection.
- ➜ There is still room for improvement.
- ➜ Chunking strategies are susceptible to versioned documents.

Thank you!