Wikipedia in the Pocket

Indexing Technology for Near-duplicate Detection and High Similarity Search

Martin Potthast
Faculty of Media, Media Systems
Bauhaus University Weimar
99421 Weimar, Germany
martin.potthast@medien.uni-weimar.de

ABSTRACT

We develop and implement a new indexing technology which allows us to use complete (and possibly very large) documents as queries, while having a retrieval performance comparable to a standard term query. Our approach aims at retrieval tasks such as near-duplicate detection and high similarity search. To demonstrate the performance of our technology we have compiled the search index "Wikipedia in the Pocket", which contains about 2 million English and German Wikipedia articles.[1] This index—along with a search interface—fits on a conventional CD (0.7 gigabyte). The ingredients of our indexing technology are similarity hashing and minimal perfect hashing.

Categories and Subject Descriptors: E.2 [Data]: Data Storage Representations—*Hash-table representations*; H.3.1 [Information Storage and Retrieval]: Content Analysis and Indexing—*Abstracting methods, Indexing methods*

General Terms: Algorithms, Performance

Keywords: hash-based indexing, fuzzy-fingerprinting, near-duplicate detection

Similarity Hashing

We use tailored similarity hash functions $h_\varphi : D \rightarrow U, U \subset \mathbf{N}$, to map documents with high pairwise similarity onto the same integer hashcode with high probability. Fuzzy-fingerprinting, which was specifically designed for text-based information retrieval, serves as a means for the construction of such hash functions [2]. It is based on the definition of a small number of k, $k \in [10, 100]$, prefix equivalence classes. A prefix class, for short, contains all terms starting with the same prefix. The computation of $h_\varphi(d)$ happens in the following steps: (1) Computation of $\mathbf{pf}$, a k-dimensional vector that quantifies the distribution of the index terms in d with respect to the prefix classes. (2) Normalization of $\mathbf{pf}$ using a corpus that provides a representative cross-section of the source language, and computation of $\Delta_{\mathbf{pf}} = (\delta_1, \ldots, \delta_k)^T$, the vector of deviations to the expected distribution.[2] (3) Fuzzification of $\Delta_{\mathbf{pf}}$ by projecting the exact deviations according to diverse fuzzification schemes.

Index Data Structure

Given a similarity hash function $h_\varphi : D \rightarrow U$, an index can be directly constructed by means of a hash table $\mathcal{T}$ along with a standard hash function $h : U \rightarrow \{1, \ldots, |\mathcal{T}|\}$; h maps the universe of hashcodes, U, onto the $|\mathcal{T}|$ storage positions. To index a document d a reference to it is added to the bucket at storage position $h(h_\varphi(d))$; likewise all documents $D' \subseteq D$ similar to d can be retrieved from this bucket.

In the case of Wikipedia in the Pocket all documents to be indexed are given and hence we can construct a minimal perfect hash function h. If h is perfect no two hashcodes will be mapped to the same storage position in $\mathcal{T}$, moreover, if h is minimal no additional storage positions are needed to ensure the former. Hence the resulting index is space optimal and allows the retrieval of similar documents in $O(1)$ time.

However, space optimal hashing entails also shortcomings: The minimal perfect hash function h has a $4.6 \cdot |D|$ byte memory overhead to represent the perfect mapping [1]. Furthermore, keys unknown prior to the construction of h will also map to a storage position in $\mathcal{T}$. Therefore the preimage of h must be stored twice, in h, as well as at the respective storage positions of $\mathcal{T}$. The former allows a perfect mapping, the latter prevents false matches. We have relaxed the latter and store small checksums: The probability of a false match between two hashcodes $x, y \in U, x \neq y$, that map to the same storage position is $p = \left(\frac{1}{2}\right)^c$ where c denotes the bit length of the checksum—in our case $c = 16$ bit.

Figure 1 shows benchmark data of the index. The table lists the sizes of the English and German index, the diagram shows precision and recall curves dependent on similarity thresholds.

Property	Value
Size (English)	359 MB
Size (German)	118 MB
Avg. postlist length	1.17 Entries

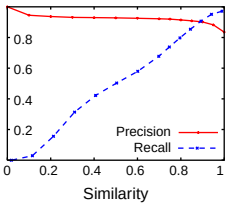

Figure 1: Benchmark data of Wikipedia in the Pocket

References

[1] F. Botelho, Y. Kohayakawa, and N. Ziviani. A Practical Minimal Perfect Hashing Method. In *Proceedings of the 4th International Workshop on efficient and Experimental Algorithms (WEA05)*, volume 3505 of *Lecture Notes in Computer Science*, pages 488–500. Springer, 2005.

[2] B. Stein. Fuzzy-Fingerprints for Text-Based Information Retrieval. In K. Tochtermann and H. Maurer, editors, *Proceedings of the 5th International Conference on Knowledge Management (I-KNOW 05), Graz*, Journal of Universal Computer Science, pages 572–579. Know-Center, July 2005.

[1] A Wikipedia snapshot from November 4th, 2006 was indexed.
[2] The British National Corpus is used as reference.

Copyright is held by the author/owner(s).
SIGIR'07, July 23–27, 2007, Amsterdam, The Netherlands.
ACM 978-1-59593-597-7/07/0007.