Tochtermann, Maurer (Eds.): Proceedings of the I-KNOW ’05, Graz
5th International Conference on Knowledge Management
Journal of Universal Computer Science, pp. 572-579, ISSN 0948-695x

Fuzzy-Fingerprints for Text-Based Information Retrieval

Benno Stein
(Bauhaus University Weimar, Germany
benno.stein@medien.uni-weimar.de)

Abstract: This paper introduces a particular form of fuzzy-fingerprints—their construction, their interpretation, and their use in the field of information retrieval. Though the concept of fingerprinting in general is not new, the way of using them within a similarity search as described here is: Instead of computing the similarity between two fingerprints in order to access the similarity between the associated objects, simply the event of a fingerprint collision is used for a similarity assessment. The main impact of this approach is the small number of comparisons necessary to conduct a similarity search.

Key Words: similarity search, information retrieval, fingerprints, locality-sensitive hashing

Category: H.3 Information Storage and Retrieval

1 Introduction

According to Merriam-Webster the term “fingerprint” has two meanings, which relate to two different aspects of identification, that is to say: authentification and recognition.

> *“Fingerprint: Something that identifies, (a) as a trait, trace, or characteristic revealing origin or responsibility, (b) as an analytical evidence (as a spectrogram) that characterizes an object or substance.”*
>
> [Merriam-Webster Collegiate Dictionary]

In fact, the paper in hand focuses on a third aspect where fingerprints can be used: As an indicator for a high similarity between the fingerprinted objects. The main line of argumentation is as follows.

Given are a set D of n objects, a similarity concept for D, and some object $d \in D$, and we are interested in finding the most similar object $d' \in D$ with respect to d. Normally, the similarity between two objects from D is not measured directly but is based on a certain abstraction or “model”. We assume that a model $\mathbf{d}$ for an object $d \in D$ is an orderd set of m metric features, which means that the objects in D refer to points in an m-dimensional vector space. Let $\mathcal{D}$ comprise the set of models for the set D of objects. The similarity between two objects $d_1, d_2 \in D$ shall be inversely proportional to the distance of their feature vectors $\mathbf{d_1}, \mathbf{d_2} \in \mathcal{D}$. The similarity is measured by a function $\varphi(\mathbf{d_1}, \mathbf{d_2})$ which maps onto $[0;1]$, with 0 and 1 indicating no and a maximum similarity respectively; φ may rely on the l_1 norm, the l_2 norm, or on the angle between the feature vectors.

Obviously the most similar object d' respecting d maximizes $\varphi(\mathbf{d}, \mathbf{d}')$ and d' can be found by a linear search. Less well known may be the results from Weber et al.: In their paper the authors give analytical and empirical evidence that finding d' cannot be done better than in linear time in $|D|$, if the dimensionality, m, of the feature space is around 10 or higher [Weber et al. 1998]. This relates in particular to all conventional space-partitioning methods like grid-files, KD-trees, or quad-trees, as well as to data-partitioning index trees such as R-tree, Rf-tree, X-tree, etc.

Observe in this connection that deciding "$d \in D$", i. e., testing whether or not d is a member in D can be done in virtually constant time, by means of hashing [see 1]. This concept can be extended towards similarity search if we had some kind of "fuzzy" hash function, $h_\varphi : D \to U$, which maps the set D of objects to a universe U of keys, $U \subset \mathbf{N}$, and which fulfills the following property:

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

Put another way, a collision that occurs between two elements from D indicates a high similarity between them. We designate $h_\varphi(d)$ as a fuzzy fingerprint of $d \in D$ at level ε. Given a function h_φ with Property (1) we can compute the universe U of fuzzy-fingerprints for a set D of objects, then map U into the slots of a hash table $T[0..|D|]$ using a hash function h, and finally store in T the references to the sets of similar objects with respect to h_φ. Based on h_φ, h, and T, a similarity search in D can be done in virtually constant time instead of in time proportional to the size of the object collection. We call this approach fuzzy-fingerprinting.
However, one is confronted with the following restrictions:

1. A fuzzy hash function h_φ has to be constructed such that h_φ resembles φ in a (small) ε-neighborhood. Note that the quality of h_φ can be measured with respect to a given model and φ—it is the portion of $\mathcal{D}$ that fulfills the reverse implication of Property (1) which reads:

$$\varphi(\mathbf{d}, \mathbf{d}') \geq 1 - |\varepsilon| \;\Rightarrow\; h_\varphi(d) = h_\varphi(d')$$

2. No similarity *order* between objects can be defined since fuzzy-fingerprinting with a function h_φ simplifies a fine-grained similarity quantification based on φ towards the binary concept "similar or not similar".

1.1 Contributions of the Paper

The considerations above are kept rather generic, and they can be applied to various search problems: D may represent a text corpus, a compilation of audio files, or an image collection, wherein similar objects for a given query have to be found. In principle, there is no restriction for fuzzy-fingerprinting if an adequate model for the objects in D along with an effective similarity function φ can be stated. We say "in principle" here, since the crucial barrier is the construction of a fuzzy hash function h_φ, which should exploit knowledge about the domain.

In this paper we will present a function h_φ to make fuzzy-fingerprinting amenable for text-based information retrieval tasks. D represents a text corpus, and we are interested in solving the following tasks:

[1] Though this claim does not hold in general, experience shows that for a given application hash functions with this property can be constructed [Cormen et al. 1990].

(a) Find (in virtually constant time) the most similar documents $D' \subset D$ for a given query document d.
(b) Find (in virtually linear time in $|D|$) all duplicate documents in D.

In detail, the contributions are: We define a family of functions h_φ to compute text document fingerprints, which consider the size of the documents. We report on experiments that show the tradeoff in time and quality between conventional and fingerprint-based text retrieval tasks.

The remainder of this paper is organized as follows. The next section gives a short overview on fingerprint methods and similarity search in information retrieval. Section 3 introduces a new hash function h_φ, and Section 4 presents results from a practical analysis, based on several text corpora and different retrieval tasks.

2 Related Work

In the context of information retrieval a fingerprint $h(d)$ can be considered as a set of substrings taken from d, which may serve to identify d uniquely. Typical and advanced fingerprinting applications include the following:

- elimination of duplicates [Chakrabarti 2003]
- elimination of near duplicates [Fetterly et al. 2003]
- retrieval of similar documents [Pereira Jr and Ziviani 2003]
- identification of source code plagiarism [Culwin et al. 2001; Prechelt et al. 2000]
- identification of versioned and plagiarized documents [Hoad and Zobel 2003; Finkel et al. 2002]

Note that the fingerprint technology as described for example in [Hoad and Zobel 2003] is used within either of two tasks: To identify two duplicate documents, d and d', by means of hashing, or to construct for a document d a compact document model $\mathbf{d}$. In the former case a hash collision $h(d) = h(d')$ is used as an indicator for identical documents, in the latter case a special type of similarity comparison between $\mathbf{d}$ and the fingerprints of all documents of a collection is made.

As stated at the outset, this is different to our intention: We address the task of a near similarity search but employ fuzzy hash functions to *avoid a pairwise similarity comparison*. There is little research that points in the same direction. Among others, the concept of locality-sensitive hashing, developed by Indyk and Motwani, counts to this [Gionis et al. 1999; Indyk 2005; Indyk and Motwani 1998]. The concept was introduced for the purposes of devising main memory algorithms for nearest neighbor search; it is not tailored to a particular domain or application but uses the vector representation of a high-dimensional search problem as its generic starting point.

In the work of Weber et al. the method of so-called vector approximation files is introduced to reduce the amount of data that must be read during similarity searches in high dimensions [Weber et al. 1998; Weber and Blott 1997]. The reported image retrieval experiments demonstrate that the method outperforms the best tree-based data structures.

3 Realization of Fuzzy-Fingerprints

We now introduce a fuzzy hash function h_φ to compute a fuzzy-fingerprint for a given document $d \in D$. As reference similarity function φ in Property 1 the well-known cosine measure along with the vector space model is employed.

While most fingerprint approaches rely on the original document d, from which substrings are selected and given to a mathematical function, our approach can be developed simplest from a document's vector space model $\mathbf{d}$. The key idea behind h_φ is an analysis and comparison of the distribution of the index terms in $\mathbf{d}$ with respect to their expected term frequency class [see 2]. In particular, we abstract the concept of term frequency classes towards *prefix frequency classes*, by comprising index terms into a small number of equivalence classes such that all terms from the same equivalence class start with a particular prefix. E. g., there might be the equivalence class of terms whose first character is from the set {"a", "A"} or, as the case may be, the equivalence class of terms whose first character is from the set {"x", "X", "y", "Y", "z", "Z"}.

Based on the analysis of extensive corpora, a standard distribution of index term frequencies can be computed, and, for a predefined set of prefixes, the a-priory probability of a term being member in a certain prefix class can be stated. The deviation of a document's term distribution from these a-priory probabilities forms a document-specific characteristic that can be encoded as a compact fingerprint.

Following this idea, a fuzzy-fingerprint $h_\varphi(d)$ for a document $d \in D$ is constructed within the following four steps; Figure 2 illustrates the procedure.

1. Extraction of the set of index terms from d, e. g. by computing its vector space model $\mathbf{d}$. In connection with Web documents this includes the removal of rendering tags and scripting code.
2. Computation of $\mathbf{pf}$, the vector of relative frequencies of k prefix classes for the index terms in $\mathbf{d}$. Our experiments rely on prefix classes that are characterized by a single alphabetic character. I. e., typical values for the number k of prefix classes are between 10 and 30.
3. Normalization of $\mathbf{pf}$ respecting the prefix classes of a reference corpus and computation of Δ_{pf}, the vector of deviations to the expected distribution. Here, the normalization grounds on the British National Corpus, which is a 100 million word collection of samples of written and spoken language from a wide range of sources, designed to represent a wide cross-section of current British English [Aston and Burnard 1998].
4. Fuzzyfication of Δ_{pf} by abstracting the exact deviations in Δ_{pf} towards diverse fuzzy deviation schemes. We propose the schemes depicted in Figure 1, which means that a deviation either falls in one of two or in one of three intervals.

[2] The term frequency class, also called word frequency class, can be used as an indicator of a word's customariness. Let $\mathcal{D}$ be a representative text corpus, let $|\mathcal{D}|$ be the number of words (terms) in $\mathcal{D}$, and let $f(w)$ denote the frequency of a word $w \in \mathcal{D}$. In accordance with [University of Leipzig 1995] the word frequency class $c(w)$ of a word $w \in \mathcal{D}$ is $\lfloor \log_2(f(w^*)/f(w)) \rfloor$, where w^* denotes the most frequently used word in $\mathcal{D}$.

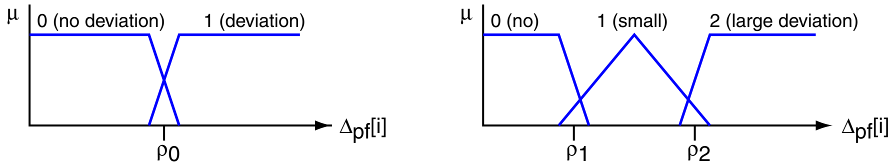

Figure 1: The two fuzzy deviation schemes that are used for the fingerprint construction.

For a fuzzification scheme ρ with r deviation intervals Formula 2 defines how a document's normalized deviation vector Δ_{pf} of length k is encoded as a fuzzy-fingerprint $h_\varphi^{(\rho)}$:

$$h_\varphi^{(\rho)}(d) = \sum_{i=0}^{k-1} \delta_i^{(\rho)} \cdot r^i, \qquad \text{with } \delta_i^{(\rho)} \in \{0, \ldots, r-1\} \tag{2}$$

$\delta_i^{(\rho)}$ is a document-specific value and encodes the fuzzified deviation of the frequency of prefix class i when applying fuzzy deviation scheme ρ. Note that in our experiments up to four variations of the parameters ρ in a fuzzification scheme are applied; $h_\varphi(d)$ can be considered as a set that comprises the fuzzy-fingerprints $h_\varphi^{(\rho)}$ for different values of ρ.

Remarks. (*i*) The granularity of the fingerprints is controlled within two dimensions at the following places: In Step 2, by the number k of equivalence classes, say, different prefix codes to be distinguished, and in Step 4, by the resolution r of the fuzzy deviation schemes. (*ii*) Since $h_\varphi(d)$ computes a *set* of fingerprints for a document d, we adapt Property 1 and agree upon the following understanding of hash collisions:

$$h_\varphi(d) \cap h_\varphi(d') \neq \emptyset \;\Rightarrow\; \varphi(\mathbf{d}, \mathbf{d}') \geq 1 - \varepsilon \tag{3}$$

(*iii*) Finally, recall that in the vector space model all information about term order is lost. Consequently, the presented fuzzy-fingerprint approach does not encode order information either.

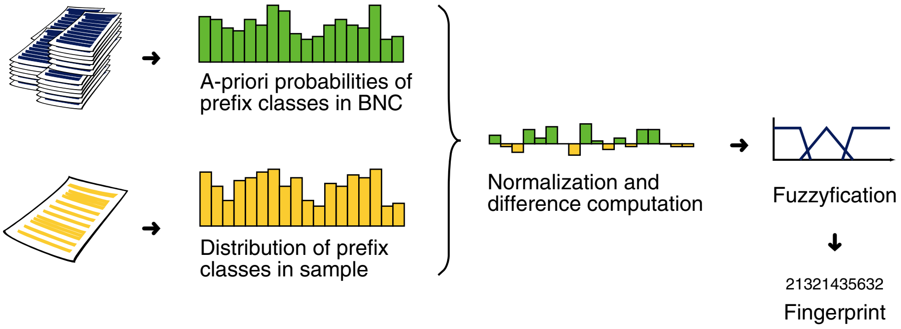
A-priori probabilities of prefix classes in BNC
Distribution of prefix classes in sample
Normalization and difference computation
Fuzzyfication
21321435632
Fingerprint

Figure 2: Pictorial overview of the fuzzy-fingerprint construction process.

4 Fuzzy-Fingerprints in a Near Similarity Application

This section presents results of an implementation of h_φ and its application within real world similarity retrieval tasks. The purpose of our experiments is twofold. Firstly, we want to shed light on the practical performance of fuzzy-fingerprints with respect to retrieval accuracy. Secondly, we want to gain evidence on the high runtime performance of fuzzy-fingerprints in comparison with traditional approaches. The parameters of h_φ are given in Table 1.

4.1 Experimental Setting

The presented results of the retrieval analysis rely on two copora. One corpus consists of all “Request For Comments” (RFCs), a collection of about 3600 text files on Internet technology [Postel 2004]. Since the documents in the RFC collection are versioned and thus include update documents, the existence of pairs of documents with a high similarity is very likely.

A second corpus is made up of about 15000 Internet documents collected with a breadth-first-search crawl starting from the “Linux Documentation Project” [Aznar 2004]. For this corpus HTML documents and text documents were downloaded, the visible portion of the HTML documents were extracted, and documents with less than 50 plain words were discarded. To ensure that solely English documents are in this corpus a stopword-based language test was applied [see 3]. Again, documents of a high similarity are likely to occur within this collection since Linux FAQs etc. are frequently updated and, in particular, mirrored on several sites.

4.2 Fuzzy-Fingerprints at Work

When using fingerprints in a retrieval process instead of a ”rich” document model, completeness cannot be guaranteed: There may be documents that are very similar to each other under, for instance, the vector space model—though their fingerprints are different. Hence, the key question here relates to the quality of recall, i. e., the fraction of similar documents that can be identified as such by means of their fuzzy-fingerprint.

[3] The test is similar to the test that has been used to compile the TREC Web Collections [Text Retrieval Conference 2003].

	Number of prefix classes k	Number of deviation intervals r	Number of fuzzification schemes ρ
h_φ	18	3	3

Table 1: Parameters of h_φ of the fuzzy-fingerprint. Three variations of the fuzzification scheme shown on the right-hand side in Figure 1 are used. The prefixes that define the equivalence classes are of length one, i. e. one class for each letter where the letters $\{j, q, u, v, w, x, y, z\}$ are discarded since their prefix frequencies in the English language is pretty low.

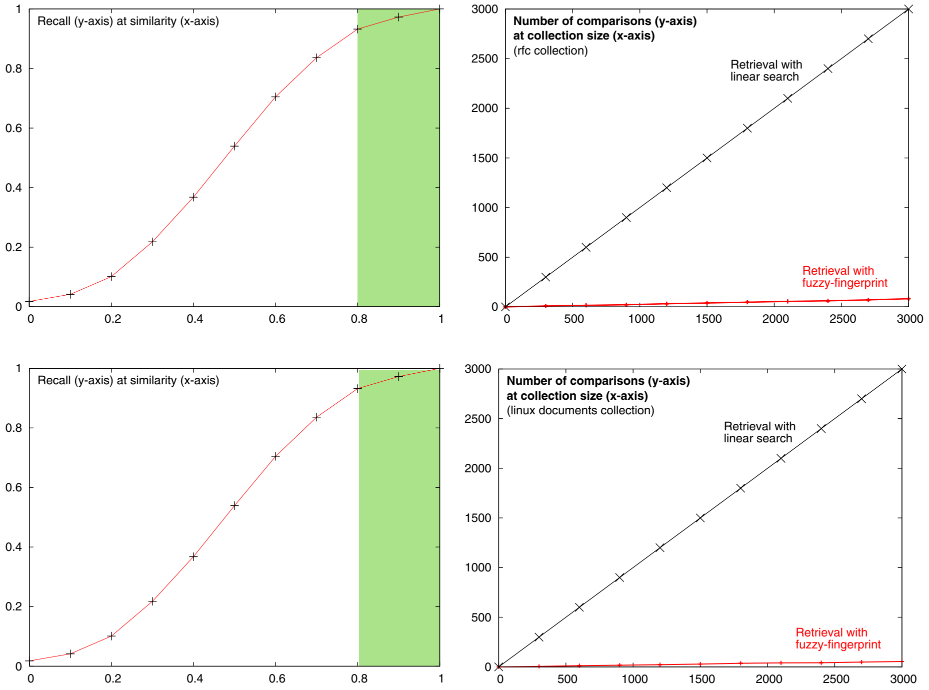

Figure 3: The plots on the left-hand side show the recall at similarity values that were achieved for the retrieval with fuzzy-fingerprints. The plots on the right-hand side illustrate the retrieval speed up: They contrast the number of comparisons of the standard retrieval process (diagonal line) and of the fuzzy-fingerprint retrieval process.

In the experiments we employed the cosine measure along with the vector space model to assess the reference similarity, and we computed the recall values with respect to different cosine similarity thresholds. The plots on the left-hand side in Figure 3 show the resulting *recall at similarity* curves, which look very promising: For the sets of documents that are similar to each other ($> 80\%$, see the shaded area) high recall-values were achieved for queries based on the fuzzy-fingerprint retrieval.

The question of *precision* reads as follows: How many documents that yield the same fuzzy-fingerprint under h_φ are actually similar to each other? Note that the documents whose fingerprints are involved in a collision form candidates for a high similarity, and a subsequent in-depth similarity analysis based on the vector space model must be applied for them. Since by a standard retrieval approach the entire collection is investigated, the ratio between the collection size and the size of the collision set can directly be interpreted as the factor for retrieval speed-up. The plots on the right-hand side in Figure 3 illustrate the sizes of both sets: The diagonal corresponds to the collection size; the line below, close to the x-axis, shows the average size of the collision set when each document in the collection is used as a query. Obviously, the use of h_φ leads to a significant retrieval speed-up.

5 Summary and Current Work

Fuzzy-fingerprints as proposed in Section 3 are an exciting research field: They can be considered as a heuristic application of the theory of locality-sensitive hashing to the area of text-retrieval. The experimental analyses show that a significant retrieval speed-up can be bought with a small (0-0.1) sacrifice in recall-quality—a tradeoff that is acceptable for many retrieval and mining scenarios.

Our current work focuses on the following open questions: (*i*) In which form can theoretical results of locality-sensitive hashing be transfered to text-specific fingerprints? (*ii*) How can the presented approach be made amenable to plagiarism analysis and chunking? Note that a meaningful comparison of large documents such as books can not achieved with a single fingerprint.

References

[Aston and Burnard 1998] Guy Aston and Lou Burnard. The BNC Handbook. `http://www.natcorp.ox.ac.uk/what/whatis.html`, 1998.

[Aznar 2004] G. Aznar. The Linux Documentation Project. `http://www.tldp.org`, 2004.

[Chakrabarti 2003] Soumen Chakrabarti. *Mining the Web*. Morgan Kaufmann, 2003.

[Cormen et al. 1990] Thomas H. Cormen, Charles E. Leiserson, and Ronald L. Rivest. *Introduction to Algorithms*. The MIT Press, Cambridge. Massachusetts, 1990.

[Culwin et al. 2001] F. Culwin, A. MacLeod, and T. Lancaster. Source Code Plagiarism in UK HE Schools—Issues, Attitudes and Tools. SBU-CISM-01-01, South Bank Univ., 2001.

[Fetterly et al. 2003] Dennis Fetterly, Mark Manasse, and Marc Najork. SOn the Evolution of Clusters of Near-Duplicate Web Pages. In *Proc. of the 1st Latin American Web Congress, LA-WEB 2003*. IEEE, 2003. ISBN 0-7695-2058-8/03.

[Finkel et al. 2002] R. A. Finkel, A. Zaslavsky, K. Monostori, and H. Schmidt. Signature Extraction for Overlap Detection in Documents. In *Proc. of the 25th Australian Conf. on Computer science*, pages 59–64. Australian Computer Soc., Inc., 2002. ISBN 0-909925-82-8.

[Gionis et al. 1999] Aristides Gionis, Piotr Indyk, and Rajeev Motwani. Similarity Search in High Dimensions via Hashing. In *Proc. of the 25th VLDB Conf. Edinburgh, Scotland*, 1999.

[Hoad and Zobel 2003] T. C. Hoad and J. Zobel. Methods for Identifying Versioned and Plagiarised Documents. *Am. Soc. f. Inf. Science and Technology*, 54(3):203–215, 2003.

[Indyk 2005] Piotr Indyk. *Handbook of Discrete and Computational Geometry*, chapter Nearest Neighbors in High-dimensional Spaces. CRC Press, 2005.

[Indyk and Motwani 1998] Piotr Indyk and Rajeev Motwani. Approximate Nearest Neighbor—Towards Removing the Curse of Dimensionality. In *Proc. of the 30th Symposium on Theory of Computing*, pages 604–613, 1998.

[Pereira Jr and Ziviani 2003] A. R. Pereira Jr and N. Ziviani. Syntactic Similarity of Web Documents. In *Proc. of the 1st Latin American Web Congress, LA-WEB 2003*. IEEE, 2003.

[Postel 2004] Jon Postel. RFC Collection. `http://www.rfc-editor.org`, 2004.

[Prechelt et al. 2000] L. Prechelt, G. Malpohl, and M. Philippsen. JPlag: Finding Plagiarisms among a Set of Programs. Report 2000-1, Univ. of Karlsruhe, Comp. Science Dept., 2000.

[Text Retrieval Conference 2003] The TREC Web Document Collection. `http://trec.nist.gov`, 2003.

[University of Leipzig 1995] Wortschatz. `http://wortschatz.uni-leipzig.de`, 1995.

[Weber and Blott 1997] Roger Weber and Stephen Blott. An Approximation-based Data Structure for Similarity Search. Report TR1997b, ETH Zentrum, Zurich, Switzerland, 1997.

[Weber et al. 1998] Roger Weber, Hans-J. Schek, and Stephen Blott. A Quantitative Analysis and Performance Study for Similarity-Search Methods in High-Dimensional Spaces. In *Proc. of the 24th VLDB Conf. New York, USA*, pages 194–205, 1998.