Dominich, Kiss (Eds.): Proceedings of the ICTIR'07, Budapest
1st International Conference on the Theory of Information Retrieval
pp. 85-93, ISSN 1587-2386, Foundation for Information Society, 2007.

# CONSTRUCTION OF COMPACT RETRIEVAL MODELS

## *Unifying Framework and Analysis*

Benno Stein and Martin Potthast
*Faculty of Media, Media Systems, Bauhaus University Weimar, 99421 Weimar, Germany*
*{benno.stein | martin.potthast}@medien.uni-weimar.de*

Keywords: document model, dimension reduction, hash-based search, fingerprinting, shingling

Abstract: In similarity search we are given a query document $d_q$ and a document collection $D$, and the task is to retrieve from $D$ the most similar documents with respect to $d_q$. For this task the vector space model, which represents a document $d$ as a vector $\mathbf{d}$, is a common starting point. Due to the high dimensionality of $\mathbf{d}$ the similarity search cannot be accelerated with space- or data-partitioning indexes; de facto, they are outperformed by a simple linear scan of the entire collection (Weber et al., 1998).

In this paper we investigate the construction of compact, low-dimensional retrieval models and present them in a unified framework. Compact retrieval models can take two fundamentally different forms: (1) As $n$-gram vectors, comparable to vector space models having a small feature set. They accelerate the linear scan of a collection while maintaining the retrieval quality as far as possible. (2) As so-called document fingerprints. Fingerprinting opens the door for sub-linear retrieval time, but comes at the price of reduced precision and incomplete recall.

We uncover the two—diametrically opposed—paradigms for the construction of compact retrieval models and explain their rationale. The presented framework is comprehensive in that it integrates all well-known construction approaches for compact retrieval models developed so far. It is unifying since it identifies, quantifies, and discusses the commonalities among these approaches. Finally, based on a large-scale study, we provide for the first time a "compact retrieval model landscape", which shows the applicability of the different kinds of compact retrieval models in terms of the rank correlation of the achieved retrieval results.

## 1 MOTIVATION

A retrieval model captures retrieval-specific aspects of a real-world document such that an information need or a retrieval task at hand can be efficiently addressed. The terminology is not used in a consistent way; in the literature also the terms "document model" and "retrieval strategy" are used (Baeza-Yates and Ribeiro-Neto, 1999; Grossman and Frieder, 2004). Note that, throughout the paper, we distinguish between the real-world document $d$ and its vector representation $\mathbf{d}$.

**Definition 1 (Retrieval Model)** *Let $D$ be a set of documents, and let $Q$ be a set of information needs or queries. A retrieval model $\mathcal{R}$ for $D$ and $Q$ is a tuple $\langle \mathbf{D}, \mathbf{Q}, \rho_\mathcal{R} \rangle$, whose elements are defined as follows:*

1. *$\mathbf{D}$ is the set of representations of the documents $D$. $\mathbf{d} \in \mathbf{D}$ may capture layout aspects, the logical structure, or semantic aspects of a document $d \in D$.*
2. *$\mathbf{Q}$ is the set of query representations or formalized information needs.*
3. *$\rho_\mathcal{R}$ is the retrieval function. It quantifies, as a real number, the relevance of a document representation $\mathbf{d} \in \mathbf{D}$ with respect to a query representation $\mathbf{q} \in \mathbf{Q}$:*

$$\rho_\mathcal{R} : \mathbf{Q} \times \mathbf{D} \to \mathbf{R}$$

We use the term "compact retrieval model" in a rather informal way, and typically in comparison to the standard vector space model, VSM. Compact retrieval models imply a much smaller representation and an improved runtime performance to address a retrieval task.

For a given query $q \in Q$ most retrieval models provide a ranking among the set of result documents $D_q$, $D_q \subset D$, that could satisfy the information need. This applies to the class of compact retrieval models whose representation is vector-based as well. Since the representation $\mathbf{d}'$ of a compact retrieval model is smaller than the standard VSM representation $\mathbf{d}$, a retrieval speed-up by the constant factor $|\mathbf{d}|/|\mathbf{d}'|$ is achieved. Under the $O$-calculus the computational effort remains linear in the collection size, say, $O(|D|)$.

However, when putting the size constraints to its extremes, the hash-based search or fingerprinting approaches come into play. They simplify a continuous similarity relation to the binary concept “similar or not similar”: By means of a multi-valued similarity hash-function $h_\varphi$, a vector-based representation $\mathbf{d}$ is mapped onto a small set of hash codes $h_\varphi(\mathbf{d})$. Two possibly high-dimensional vectors, $\mathbf{d}_1, \mathbf{d}_2$, are considered as similar if their fingerprint representations share some hash code:

$$\big(h_\varphi(\mathbf{d}_1) \cap h_\varphi(\mathbf{d}_2)\big) \neq \emptyset \quad \Rightarrow \quad \varphi(\mathbf{d}_1, \mathbf{d}_2) \geq 1 - \varepsilon$$

With fingerprinting a sub-linear retrieval time can be achieved; in fact, the retrieval time is in $O(|D_q|)$. Because of the exponential similarity distribution in a collection $D$, the result set $D_q$ for a query $q$ increases logarithmically in the collection size $|D|$, if $h_\varphi$ is properly designed. In practical applications $|D_q|$ can hence be assessed with a constant.

## 1.1 Use Cases

The main reason for the use of compact retrieval models is retrieval speed up; a second reason may be the small memory footprint. Since there is no free lunch one pays for these benefits, whereas the price depends on the use case. Table 1 lists the use cases where compact retrieval models have proven to be effective. Note that the table distinguishes between vector representations and fingerprint representations. The shown assessments (– / inappropriate, o / acceptable, + / appropriate, ++ / ideal) take the particular precision and recall requirements of the use cases into account.

High similarity search is also known as near-duplicate detection, where the task is to find in a collection $D$ all documents whose pairwise similarity is close to 1. The use case “similarity search” pertains to standard retrieval tasks where a query $q$ or a query document $d_q$ is given, and one is interested in a result set $D_q \subset D$ of relevant documents with respect to the query. Plagiarism analysis compares to a high similarity search that is done at the paragraph level: given a candidate document $d$ the task is to find all documents in $D$ that contain nearly identical passages from $d$. Finally, post retrieval clustering is a special form of result set preparation. It plays an important role in connection with user interaction and search interfaces, if a large result set $D_q$ needs to be abstracted, categorized, or visually prepared.

| | Suitability of compact retrieval model | |
|---|---|---|
| **Use case** | **Vector** | **Fingerprint** |
| High similarity search | + | ++ |
| Similarity search | + | o |
| Plagiarism analysis | o | + |
| Post retrieval clustering | + | o |

Table 1: Use cases where compact retrieval models are successfully applied. Under the viewpoint of retrieval quality the vector representation is superior to the fingerprint representation, under the viewpoint of retrieval runtime it is vice versa.

## 1.2 Contributions

The contributions of our work are twofold, comprising conceptual and empirical results.

1. First, we organize existing research concerned with compact retrieval models and point out underlying rationales. These insights may help the developer of information retrieval solutions to select among existing technology as well as to develop new retrieval models.
2. Second, based on a large scale analysis, we analyze the practical retrieval performance of well-known representatives of compact retrieval models. The results provide a guideline for practical applications; in essence, they reflect the theoretical considerations.

# 2 CONSTRUCTION OF COMPACT RETRIEVAL MODELS

This section introduces both a unifying framework and the underlying principles for a wide range of compact retrieval models. We start with a comparison of the orders of magnitude that are typical for the use cases listed in Table 1. Subsection 2.1, which may be the most insightful contribution of this section, explains the two fundamental construction paradigms and discusses their implications. Subsection 2.2 surveys construction approaches developed so far. For overview purposes Figure 1 shows the complete construction process as UML activity diagram: starting point is a token sequence generated from a document $d$, followed by a chunking step that generates

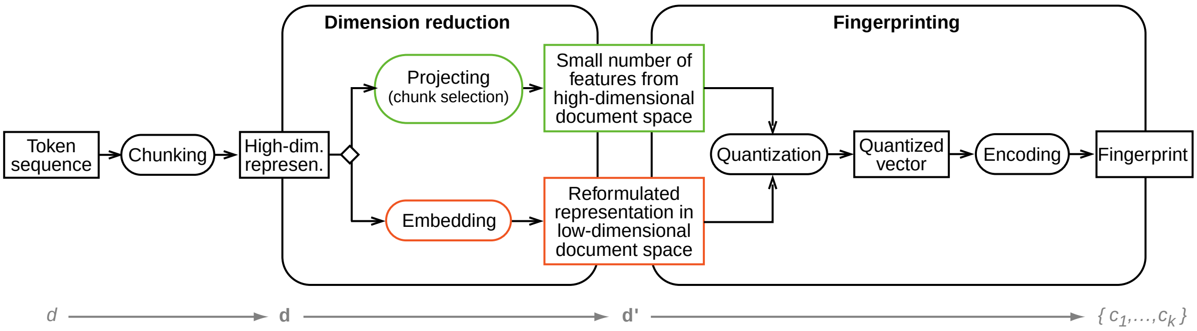
Dimension reduction
Fingerprinting
Token sequence
Chunking
High-dim. represen.
Projecting (chunk selection)
Embedding
Small number of features from high-dimensional document space
Reformulated representation in low-dimensional document space
Quantization
Quantized vector
Encoding
Fingerprint
d
d
d'
{c1,...,ck}

Figure 1: The construction of a compact retrieval model starts with a token sequence generated from a document $d$. After the dimension reduction step we are given a document representation $\mathbf{d}'$ with a small number of features, which can be used in the standard sequential retrieval process. If one is interested in binary retrieval a fingerprint can be computed from $\mathbf{d}'$.

a set of $n$-grams (= sequence of $n$ contiguous words), which form the high-dimensional document representation $\mathbf{d}$.

The challenges that must be met when constructing compact retrieval models arise from the dimensionality of the document space. Technically speaking, the document space is an inner product space.[1] The dimension $m$ of the document space depends on the chunking step which defines the underlying dictionary, $T$. $T$ is the union set of all descriptors (terms or 1-grams, 2-grams, $n$-grams) that occur in at least one document representation $\mathbf{d}$. Note that $\mathbf{d}$ may be understood as an $m$-dimensional vector whose components represent descriptors from which only a fraction has a non-zero weight. On the other hand, $\mathbf{d}$ may be understood as a list of $(\mathit{descriptor}, \mathit{weight})$-tuples, comprising only those dimensions having a non-zero weight. Under the former view the similarity between two document representations $\mathbf{d}_1$ and $\mathbf{d}_2$ can be computed with the scalar product; under the latter view a set-based measure like the Jacquard coefficient can be applied. In essence both representations are of equal power and complexity. In Figure 2, which contrasts the dimensions of different document spaces with the respective sizes of the document representations $\mathbf{d}$, a set-based representation of $\mathbf{d}$ is assumed.

Table 2 lists the exact values of different dictionaries of Wikipedia, using the the entire English collec-

[1] This is not a basic necessity but common practice in information retrieval. Exceptions include cluster-based retrieval systems, suffix-tree-based document representations, and the like.

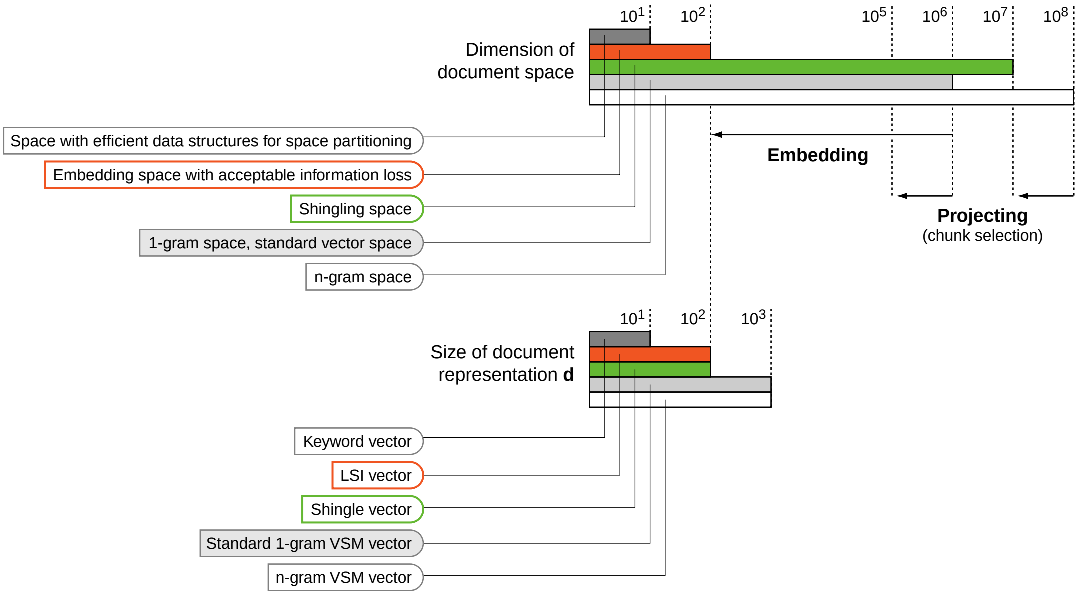

Figure 2: The diagram contrasts the dimensions of different document spaces with the sizes of the document representations. Note the logarithmic scale of the orders of magnitudes.

| Dictionary | Number of dimensions |
|---|---:|
| 1-gram space | 3 921 588 |
| 4-gram space | 274 101 016 |
| 8-gram space | 373 795 734 |
| Shingling space | 75 659 644 |

Table 2: Dictionary dimensions of the English Wikipedia collection from November 2006.

tion from November 2006 as basis. The *n*-gram space is spanned by the set of all *n*-grams in the collection, hence the 1-gram space is the standard vector space. The shingling space is spanned by the union set of all shingles (= 8-grams) that have been selected during the construction process of the documents' low-dimensional shingling representations (Broder, 2000).

Figure 2 also hints the two construction paradigms for compact retrieval models: projecting and embedding. Both aim at dimension reduction, and both provide a means for constructing a small document representation. The rationale behind projecting is a hypothesis test, whereas the rationale behind embedding is to capture as much as possible from the information of the high-dimensional representation. The next subsection discusses the implications.

## 2.1 Hypothesis Test or Model Fidelity?

Let $\mathbf{D}$ be the set of representations of the documents in a collection $D$, where each document vector $\mathbf{d} \in \mathbf{D}$ is based on 8-grams taken from its associated document $d$. Let $\mathbf{R}_\theta \subset \mathbf{D} \times \mathbf{D}$ be the set of all pairs of document vectors, $\{\mathbf{d}_1, \mathbf{d}_2\}$, whose similarity $\varphi(\mathbf{d}_1, \mathbf{d}_2)$ is above $\theta$, with $\theta \in [0.8; 1.0)$. Likewise, let $\mathbf{R}_{<\theta}$ be the set of all remaining pairs with $\varphi(\mathbf{d}_1, \mathbf{d}_2) < \theta$. Note that $|\mathbf{R}_\theta| / |\mathbf{R}_{<\theta}| \ll 1$.

Consider now two document vectors, $\mathbf{d}_1, \mathbf{d}_2 \in \mathbf{D}$, and let there be an 8-gram, $s$, that is shared among them, i. e., $s \in \mathbf{d}_1 \cap \mathbf{d}_2$. Then the question is, which of the following hypotheses shall be accepted, which shall be rejected?

$$H_0: \text{ “}\{\mathbf{d}_1, \mathbf{d}_2\} \text{ is from } \mathbf{R}_{<\theta}\text{”}$$

$$H_1: \text{ “}\{\mathbf{d}_1, \mathbf{d}_2\} \text{ is from } \mathbf{R}_{\theta}\text{”}$$

To answer this question we have to investigate the sizes of the sets $\mathbf{R}_\theta$ and $\mathbf{R}_{<\theta}$ in connection with the probability of the event that two document vectors share an *n*-gram. The sizes $|\mathbf{R}_\theta|$ and $|\mathbf{R}_{<\theta}|$ will basically follow the characteristic of the 1-gram similarity distribution shown in Figure 3, but still be more extreme because of the 8-gram representation. Typical order of magnitudes are $0.01^2 \cdot |D|^2/2$ for $|\mathbf{R}_\theta|$ and $|D|^2/2$ for $|\mathbf{R}_{<\theta}|$.

The probability $P_s$ of the event that two document vectors $\mathbf{d}_1$ and $\mathbf{d}_2$ share an *n*-gram $s$ is determined by their similarity, $\varphi(\mathbf{d}_1, \mathbf{d}_2)$, and the underlying *n*-gram length. For the two events “$\{\mathbf{d}_1, \mathbf{d}_2\} \in \mathbf{R}_\theta$” and “$\{\mathbf{d}_1, \mathbf{d}_2\} \in \mathbf{R}_{<\theta}$” Figure 4 shows the characteristic curves for $P_s$, dependent on the *n*-gram length. Altogether, the probabilities $P_0$ and $P_1$ for the events formulated as hypothesis $H_0$ and $H_1$ derive from the following relation:

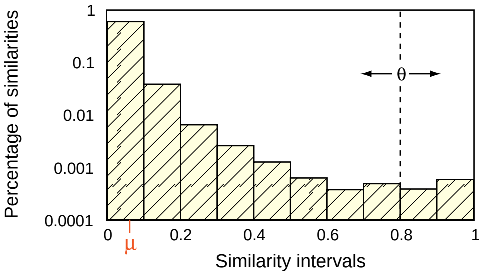

Figure 3: Similarity distribution in the Reuters Corpus Volume 1, RCV1, (Rose et al., 2002) under the 1-gram document representation (= VSM).

$$\frac{|\mathbf{R}_{<\theta}| \cdot P_s(\{\mathbf{d}_1,\mathbf{d}_2\} \in \mathbf{R}_{<\theta},\ n{=}8)}{|\mathbf{R}_\theta| \cdot P_s(\{\mathbf{d}_1,\mathbf{d}_2\} \in \mathbf{R}_\theta,\ n{=}8)} \quad \sim \quad \frac{P_0}{P_1}$$

For the Wikipedia collection we have assessed values for $P_0$ and $P_1$ assuming different similarity thresholds $\theta$. It turns out that for $\theta$-values from the interval $[0.8; 1.0)$ the probability $P_1$ is about 20 times higher than the probability $P_0$. I. e., $H_1$ is accepted, and $H_0$ is rejected.

*Remarks.* The outlined connections form the rationale of shingling (in particular) and projecting (in general). They also show that this paradigm is strongly biased towards high similarity relations, and that it cannot be applied to reason about medium similarities. The analyses of Section 3 will approve this argumentation. A diametrically opposed paradigm is embedding, which aims at model fidelity, i. e., the preservation of a wide range of similarity relations. Speaking technically, projecting relates to feature selection while embedding relates to feature reformulation.

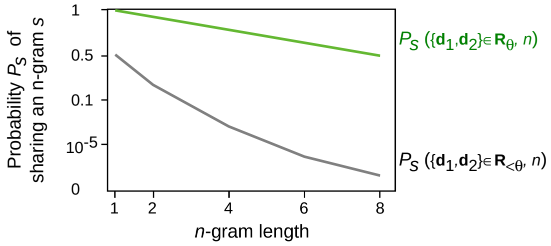

Figure 4: Probability of the event that two document vectors share an *n*-gram, dependent on *n*. The upper curve relates to highly similar pairs drawn from $\mathbf{R}_\theta$, $\theta \in [0.8; 1.0)$; the lower curve relates to pairs drawn from $\mathbf{R}_{<\theta}$. Consider in this connection the distribution of the similarities under the 1-gram document representation in Figure 3.

| **Projecting algorithm** | **(Author)** | **Characteristics of d′** |
|---|---|---|
| rare chunks | (Heintze, 1996) | the descriptors in $\mathbf{d}'$ occur at most once in $D$ |
| SPEX | (Bernstein and Zobel, 2004) | the descriptors in $\mathbf{d}'$ occur at least twice in $D$ |
| I-Match | (Chowdhury et al., 2002)<br>(Conrad et al., 2003; Kołcz et al., 2004) | $\mathbf{d}'$ contains the most discriminant terms from $\mathbf{d}$ |
| random | (misc.) | the descriptors in $\mathbf{d}'$ are randomly chosen from $\mathbf{d}$, or |
| shingling | (Broder, 2000) | the descriptors in $\mathbf{d}'$ minimize a random function over the descriptors of $\mathbf{d}$ |
| prefix anchor | (Manber, 1994) | the descriptors in $\mathbf{d}'$ start with a particular prefix, or |
| | (Heintze, 1996) | the descriptors in $\mathbf{d}'$ start with a prefix which is infrequent in $d$ |
| hashed breakpoints | (Manber, 1994) | the last byte of the descriptors in $\mathbf{d}'$ is 0, or |
| | (Brin et al., 1995) | the last word's hash value of the descriptors in $\mathbf{d}'$ is 0 |
| sliding window | (misc.) | the descriptors in $\mathbf{d}'$ start at a word $i$ mod $m$ in $d$, $m \in \{1, \ldots, \|d\|\}$, or |
| winnowing | (Schleimer et al., 2003) | the descriptors in $\mathbf{d}'$ minimize a hash function of a window sliding over $d$ |

Table 3: Summary of projecting algorithms. The rows contain the name of the construction algorithm, the authors, and a characterization of the constraints that must be fulfilled by the descriptors in $\mathbf{d}'$.

## 2.2 Construction Approaches

A basic step in all construction approaches for compact retrieval models is dimension reduction, which computes from a high-dimensional document representation $\mathbf{d}$ a representation $\mathbf{d}'$ with a small number of features (recall Figure 1).

Figure 5 organizes in a taxonomy the dimension reduction techniques that have been applied for retrieval model construction, whereas Table 3 and Table 4 provide for a short characterization of the techniques.

*Collection-specific versus Document-specific.* This distinction pays tribute to the fact that a dimension reduction approach relies either on a single document at a time (= document-specific) or on the entire collection $\mathbf{D}$ (= collection-specific). Typically the former is much more efficient with respect to runtime, while the latter enables one to integrate global considerations as well as knowledge from the retrieval task. Note, however, that a document-specific dimension reduction presumes a closed retrieval situation (Stein, 2007).

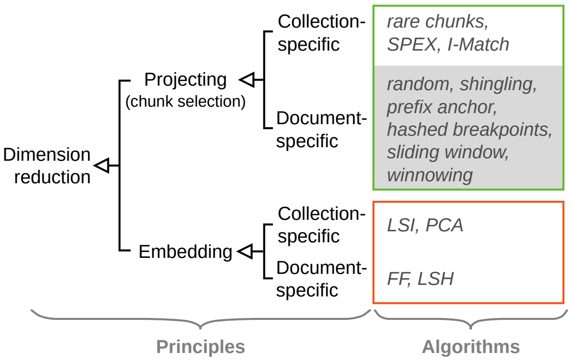

Figure 5: Survey of dimension reduction principles and algorithms. Existing surveys are restricted to document-specific algorithms that base on projecting (shown shaded).

If the use case allows the abstraction of the continuous similarity relation to a binary relation, or if retrieval time has top priority, a fingerprint can be computed from $\mathbf{d}'$. Fingerprinting applies to $\mathbf{d}'$ an additional quantization and encoding step (see again Figure 1). Quantization is the mapping of the real valued vector components to integer values. Encoding is the computation of an integer number according to some rule such as the one defined by the $l_1$-norm of the embedded and quantized $\mathbf{d}$.

Within the embedding LSH technology the quantization step is operationalized as follows. The real number line is divided into equidistant intervals each

| **Embedding algorithm** | **(Author)** | **Characteristics of d′** |
|---|---|---|
| LSI | (Deerwester et al., 1990) | $\mathbf{d}'$ is computed from the SVD of the term-document matrix |
| PCA | (Jolliffe, 1996) | $\mathbf{d}'$ is computed from the SVD of the covariance matrix of the term-document matrix |
| PLSI | (Hofmann, 2001) | $\mathbf{d}'$ contains the hidden variables of a statistical language model, computed by an EM |
| embedding LSH | (Datar et al., 2004) | the components in $\mathbf{d}'$ are the scalar products of $\mathbf{d}$ with a random vector set |
| fuzzy-fingerprinting | (Stein, 2005) | the components in $\mathbf{d}'$ are the normalized expected deviations of particular index term distributions in $d$ |

Table 4: Summary of embedding algorithms. The rows contain the name of the construction algorithm, the authors, and a characterization of the construction method responsible for the computation of $\mathbf{d}'$.

of which having assigned a unique natural number, and the components of $\mathbf{d}'$ are identified with the number of their enclosing interval. Encoding can happen in different ways and is typically done by summation (Charikar, 2002; Datar et al., 2004).

Within the fuzzy-fingerprinting technology the quantization step is achieved by applying different fuzzification schemes to the components of $\mathbf{d}'$. Encoding is done by computing the smallest number in a certain radix notation from the fuzzified deviations (Stein, 2005).

# 3 MODEL LANDSCAPE

Which is the best-suited compact retrieval model for a given task? Though this question may not be answered in its generality, a comprehensive experimental analysis and its graphical presentation can help us to understand the strong and weak points of a model. This is the idea of our model landscape, shown in Table 6, from which Table 5 shows a minimized version for orientation purposes.

The model landscape combines three important approaches, namely shingling, LSH, and fuzzy-fingerprinting (x-axis) along with four different document model sizes (y-axis): $|\mathbf{d}'|$ = 100, 50 and 10 in the first, second, and third row respectively, while in the fourth row the most compact model in the form of the related fingerprint technology is used (cf. Table 5). Each cell in Table 6 shows a histogram which quantifies the achieved retrieval performance as rank correlation value. Baseline for the rank correlation is the standard VSM model: for each combination shown in the table the rank correlation values of 1000 query results are averaged. Since we expect that compact retrieval models are biased with respect to certain similarity intervals, the rank correlation values were broken down to the following six similarity thresholds: 0, 0.25, 0.5, 0.65, 0.8, and 0.9.

As document collection for the rank correlation analysis served the Reuters Corpus Volume 1 (Rose et al., 2002). The corpus consists of Reuters news articles which have been manually assigned to categories. For each query document 1000 comparison documents were chosen from the same category. For the analysis of the fingerprinting methods the Wikipedia Revision Corpus was used. The corpus contains the complete revision history of each Wikipedia article, and hence it forms a rich source of similar document pairs. Such kind of "biased" corpus is necessary to reliably measure the recall performance of fingerprinting methods; in standard corpora the number of document pairs with a high similarity is extremely small compared to those with a low similarity. Altogether, about 7 Million document pairs from the corpus were analyzed.

| Document model size | Shingling | Embedding LSH | Fuzzy-fingerprinting |
|---|---|---|---|
| large | Shingle vector size: 100 shingles | LSH vector 100-dimensional | Prefix class vector 100-dimensional |
|  | Shingle vector size: 50 shingles | LSH vector 50-dimensional | Prefix class vector 50-dimensional |
|  | Shingle vector size: 10 shingles | LSH vector 10-dimensional | Prefix class vector 10-dimensional |
| small | Supershingling 2 × 8 Byte | LSH fingerprint 2 × 8 Byte | Fuzzy-fingerprint 2 × 8 Byte |

Table 5: Legend of the landscape of compact retrieval models shown in Table 6.

## 3.1 Rank Correlation

Let $\mathbf{D}$ be a set of document vectors, $\mathbf{d}$ a single document vector, and $\varphi$ a similarity measure. A ranking of $\mathbf{D}$ wrt. $\mathbf{d}$ under $\varphi$ is a list of all document vectors in $\mathbf{D}$, sorted in ascending order according to their similarity to $\mathbf{d}$. Let $\mathbf{D}'$ be the set of compact document vectors derived from $\mathbf{D}$, and let $\mathbf{d}'$ be the compact vector derived from $\mathbf{d}$. The correlation between the ranking of $\mathbf{D}$ wrt. $\mathbf{d}$ and the ranking of $\mathbf{D}'$ wrt. $\mathbf{d}'$ is referred to as rank correlation.

The degree of correlation between two rankings is measured with a rank correlation coefficient, yielding a value from $[-1,1]$, where $-1$ indicates a perfect anti-correlation while 1 indicates a perfect correlation. Two rank correlation coefficients have been proposed and can be used for our task, namely Spearman's $\rho$ and Kendall's $\tau$ (Kendall and Stuart, 1979; Wackerly et al., 2001):

- *Spearman's* $\rho$. The Spearman coefficient accounts the squared difference of the ranks of the $i$th document when it is represented under the standard VSM and as compact vector:

$$\rho = 1 - \frac{6 \cdot \sum_{i=0}^{|\mathbf{D}|}(r(\mathbf{d}_i,\mathbf{D}) - r(\mathbf{d}'_i,\mathbf{D}')^2}{|\mathbf{D}| \cdot (|\mathbf{D}|^2 - 1)},$$

  where $r$ denotes a ranking function which maps $\mathbf{d}_i \in \mathbf{D}$ ($\mathbf{d}'_i \in \mathbf{D}'$) to it's rank in the ranking of $\mathbf{D}$ ($\mathbf{D}'$) for a particular $\mathbf{d}$ ($\mathbf{d}'$). A significance test for $\rho$ can be done for any $|\mathbf{D}| > 30$, based on Student's $t$-distribution.

- *Kendall's* $\tau$. The Kendall coefficient compares the ranking of each pair of documents when they are represented under the standard VSM and as compact vectors. If the ranking of a document pair is

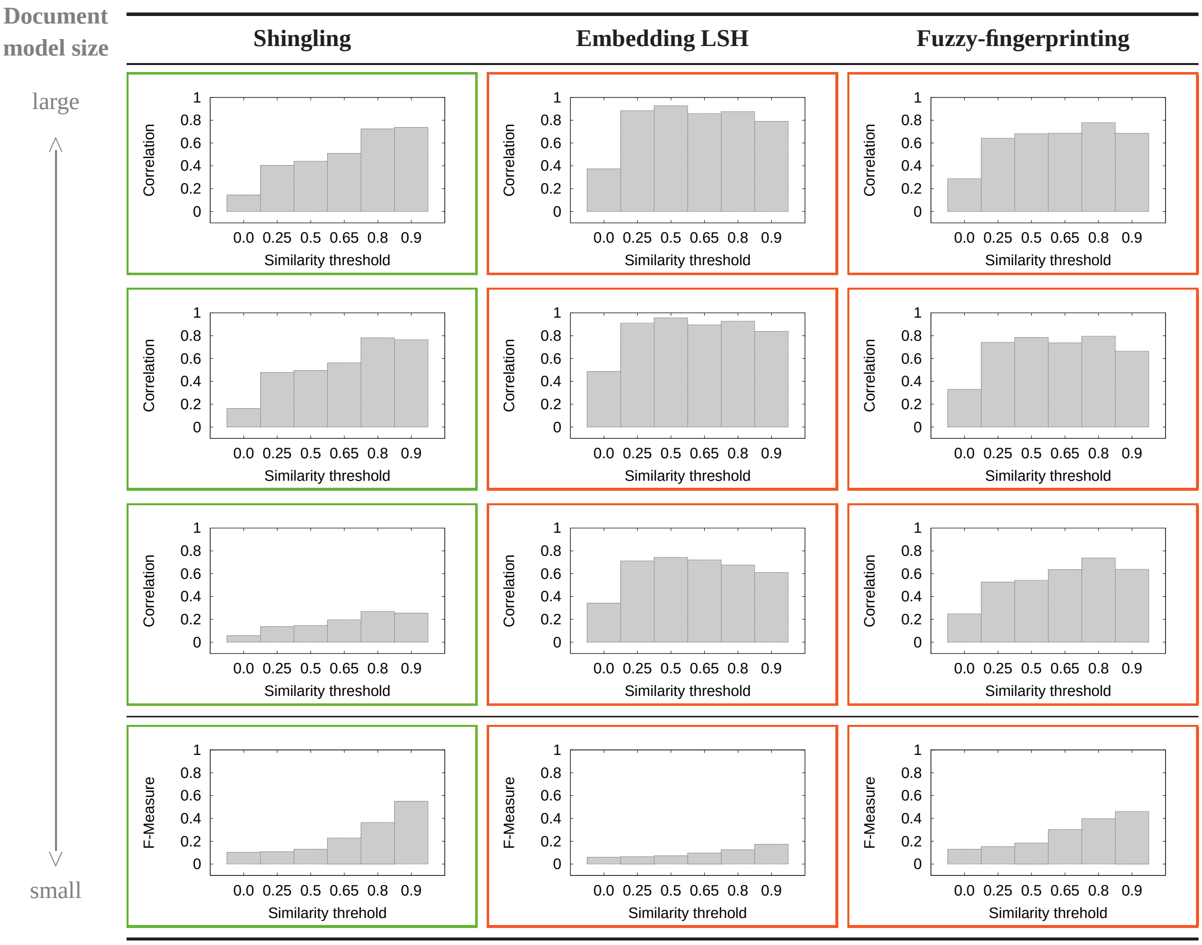

Table 6: Landscape of compact retrieval models. The document representation in the first three rows is vector-based, with $|\mathbf{d}'|$ = 100, 50 and 10 in the first, second, and third row respectively. Underlying the fourth row is the fingerprint representation of the respective approach, i. e. a small number of codes $\{c_1, \ldots, c_k\}$. Each table cell shows a histogram which quantifies the achieved retrieval performance as rank correlation value, broken down to six similarity thresholds. Baseline for the rank correlation were 1000 queries per similarity interval conducted under the standard VSM.

the same under both models they are considered as concordant.

$$\tau = 1 - \frac{2 \cdot P}{|\mathbf{D}| \cdot (|\mathbf{D}| - 1)},$$

where $P$ denotes the number of concordant document pairs. A significance test for $\tau$ can be done for any $|\mathbf{D}| > 10$, based on the normal distribution.

## 3.2 Discussion

The presented model landscape provides a comprehensive view on the characteristics of compact retrieval models. The important observations can be summarized as follows:

- An increase in the similarity threshold goes along with an increase in the rank correlation. Shingling performs worst; the rank correlation at medium similarity thresholds is considerable smaller than those of the other models. Both embedding LSH and fuzzy-fingerprinting show a high rank correlation, with a slight advantage to the former.
- A reduction in the dimensionality impairs the rank correlation for all approaches. Here, the compact models based on embedding are affected by at most 25% (cf. embedding LSH) whereas shingling decreases by more than 75%. Considering the third row of Table 6 both embedding LSH and fuzzy-fingerprinting perform similar.
- The last row of Table 6 unveils an interesting characteristic of the related fingerprints: the values of shingling and fuzzy-fingerprinting at high similarity thresholds compete with each other, while embedding LSH is about 65% behind. I. e., the good

|  | Suitability of compact retrieval model |  |  |  |  |  |
|---|---|---|---|---|---|---|
|  | Shingling |  | Embedding LSH |  | Fuzzy-fingerprinting |  |
| Use case | Vector | Fingerprint | Vector | Fingerprint | Vector | Fingerprint |
| High similarity search | ++ | ++ | ++ | ++ | ++ | ++ |
| Similarity search | – | – | o | o | o | o |
| Plagiarism analysis | + | + | o | o | + | + |
| Post retrieval clustering | – | – | o | o | + | + |

Table 7: Verbose version of the use cases from Table 1: while all compact retrieval models do a good job in high similarity search, most of them fail in connection with standard similarity search, plagiarism analysis, or post retrieval clustering.

rank correlation of embedding LSH does not imply a good retrieval performance for the related fingerprint—and vice versa, as can be seen in the case of shingling.

Table 7 gives a qualitative survey of our observations; it contrasts the suitability of the evaluated compact retrieval models for the use cases discussed at the outset.

## REFERENCES

Baeza-Yates, R. and Ribeiro-Neto, B. (1999). *Modern Information Retrieval*. Addison Wesley.

Bernstein, Y. and Zobel, J. (2004). A scalable system for identifying co-derivative documents. In Apostolico, A. and Melucci, M., editors, *Proceedings of the String Processing and Information Retrieval Symposium (SPIRE)*, pages 55–67, Padova, Italy. Springer. Published as LNCS 3246.

Brin, S., Davis, J., and Garcia-Molina, H. (1995). Copy detection mechanisms for digital documents. In *SIGMOD '95*, pages 398–409, New York, NY, USA. ACM Press.

Broder, A. Z. (2000). Identifying and filtering near-duplicate documents. In *COM'00: Proceedings of the 11th Annual Symposium on Combinatorial Pattern Matching*, pages 1–10, London, UK. Springer-Verlag.

Charikar, M. S. (2002). Similarity Estimation Techniques from Rounding Algorithms. In *STOC '02: Proceedings of the thirty-fourth annual ACM symposium on Theory of computing*, pages 380–388, New York, NY, USA. ACM Press.

Chowdhury, A., Frieder, O., Grossman, D., and McCabe, M. (2002). Collection statistics for fast duplicate document detection. *ACM Trans. Inf. Syst.*, 20(2):171–191.

Conrad, J., Guo, X., and Schriber, C. (2003). Online duplicate document detection: signature reliability in a dynamic retrieval environment. In *Proceedings of the 2003 ACM CIKM International Conference on Information and Knowledge Management (CIKM)*, pages 443–452. ACM.

Datar, M., Immorlica, N., Indyk, P., and Mirrokni, V. S. (2004). Locality-Sensitive Hashing Scheme Based on p-Stable Distributions. In *SCG '04: Proceedings of the twentieth annual symposium on Computational geometry*, pages 253–262, New York, NY, USA. ACM Press.

Deerwester, S., Dumais, S., Landauer, T., Furnas, G., and Harshman, R. (1990). Indexing by Latent Semantic Analysis. *Journal of the American Society of Information Science*, 41(6):391–407.

Grossman, D. and Frieder, O. (2004). *Information Retrieval*. Springer, second edition.

Heintze, N. (1996). Scalable document fingerprinting. In *Proceedings of the Second USENIX Electronic Commerce Workshop*, pages 191–200.

Hofmann, T. (2001). Unsupervised Learning by Probabilistic Latent Semantic Analysis. *Machine Learning*, 42:177–196.

Jolliffe, I. (1996). *Principal Component Analysis*. Springer.

Kendall, M. and Stuart, A. (1979). *The Advanced Theory of Statistics*. Hafner Press.

Kołcz, A., Chowdhury, A., and Alspector, J. (2004). Improved robustness of signature-based near-replica detection via lexicon randomization. In *KDD '04: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 605–610, New York, NY, USA. ACM Press.

Manber, U. (1994). Finding similar files in a large file system. In *Proceedings of the USENIX Winter 1994 Technical Conference*, pages 1–10, San Fransisco, CA, USA.

Rose, T., Stevenson, M., and Whitehead, M. (2002). The Reuters Corpus Volume 1 - From Yesterday's News to Tomorrow's Language Resources. In *Proceedings of the Third International Conference on Language Resources and Evaluation*.

Schleimer, S., Wilkerson, D., and Aiken, A. (2003). Winnowing: local algorithms for document fingerprinting. In *SIGMOD '03: Proceedings of the 2003 ACM SIGMOD international conference on Management of data*, pages 76–85, New York, NY, USA. ACM Press.

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

Stein, B. (2007). Principles of hash-based text retrieval. In Clarke, C., Fuhr, N., Kando, N., Kraaij, W., and de Vries, A., editors, *30th Annual International ACM SIGIR Conference*, pages 527–534. ACM.

Wackerly, D., Mendenhall III, W., and Scheaffer, R. (2001). *Mathematical Statistics with Applications*. Duxbury Press.

Weber, R., Schek, H., and Blott, S. (1998). A Quantitative Analysis and Performance Study for Similarity-Search Methods in High-Dimensional Spaces. In *Proceedings of the 24th VLDB Conference New York, USA*, pages 194–205.