Spiliopoulou et al. (Eds.): From Data and Information Analysis to Knowledge Engineering
Selected Papers from the 29th Annual Conference of the German Classification Society (GfKl)
Magdeburg, ISBN 1431-8814, pp. 430-437, ©Springer 2006.

Near Similarity Search and Plagiarism Analysis

Benno Stein[1] and Sven Meyer zu Eissen[2]

[1] Faculty of Media, Media Systems
Bauhaus University Weimar, 99421 Weimar, Germany
[2] Faculty of Computer Science,
Paderborn University, 33098 Paderborn, Germany

Abstract. Existing methods to text plagiarism analysis mainly base on "chunking", a process of grouping a text into meaningful units each of which gets encoded by an integer number. Together theses numbers form a document's signature or fingerprint. An overlap of two documents' fingerprints indicate a possibly plagiarized text passage. Most approaches use MD5 hashes to construct fingerprints, which is bound up with two problems: (*i*) it is computationally expensive, (*ii*) a small chunk size must be chosen to identify matching passages, which additionally increases the effort for fingerprint computation, fingerprint comparison, and fingerprint storage.

This paper proposes a new class of fingerprints that can be considered as an abstraction of the classical vector space model. These fingerprints operationalize the concept of "near similarity" and enable one to quickly identify candidate passages for plagiarism. Experiments show that a plagiarism analysis based on our fingerprints leads to a speed-up by a factor of five and higher—without compromising the recall performance.

1 Plagiarism Analysis

Plagiarism is the act of claiming to be the author of material that someone else actually wrote (Encyclopædia Britannica, 2005). This definition relates to text documents, which is also the focus of this paper. Clearly, a question of central importance is to what extent such and similar tasks can be automated. Several techniques for plagiarism analysis have been proposed in the past—most of them rely on one of the following ideas.

Substring Matching. Substring matching approaches try to identify maximal matches in pairs of strings, which then are used as plagiarism indicators (Gusfield (1997)). Typically, the substrings are represented in suffix trees, and graph-based measures are employed to capture the fraction of the plagiarized sections (Baker (1993); Monostori et al. (2002, 2000)). However, Finkel et al. (2002) as well as Baker (1993) propose the use of text compression algorithms to identify matches.

Keyword Similarity. The idea here is to extract and to weight topic-identifying keywords from a document and to compare them to the keywords of other documents. If the similarity exceeds a threshold, the candidate documents

are divided into smaller pieces, which then are compared recursively (Si et al. (1997); Fullam and Park (2002)). Note that this approach assumes that plagiarism usually happens in topically similar documents.
Fingerprint Analysis. The most popular approach to plagiarism analysis is the detection of overlapping text sequences by means of fingerprinting: Documents are partitioned into term sequences, called chunks, from which digital digests are computed that form the document's fingerprint. When the digests are inserted into a hashtable, collisions indicate matching sequences. Recent work that describes details and variants of this approach include Brin et al. (1995); Shivakumar and Garcia-Molina (1996); Finkel et al. (2002).

1.1 Contributions of this Paper

The overall contribution of this paper relates to the usage of fuzzy-fingerprints as an effective tool for plagiarism analysis. To understand different intentions for similarity search and plagiarism analysis we first introduce the distinction of local and global similarity. In fact, fuzzy-fingerprints can be understood as a combination of both paradigms, where the parameter "chunk size" controls the degree of locality. In particular, we use this distinction to develop a taxonomy of methods for plagiarism analysis. These considerations are presented in the following section. Section 3 reports on experiments that quantify interesting properties of our approach.

2 Fingerprinting, Similarity, and Plagiarism Analysis

In the context of information retrieval a fingerprint $h(d)$ of a document d can be considered as a set of encoded substrings taken from d, which serve to identify d uniquely.[1] Following Hoad and Zobel (2003), the process of creating a fingerprint comprises four areas that need consideration.

1. *Substring Selection.* From the original document substrings (chunks) are extracted according to some selection strategy. Such a strategy may consider positional, frequency-based, or structural information.
2. *Substring Number.* The substring number defines the fingerprint resolution. There is an obvious trade-off between fingerprint quality, processing effort, and storage requirements, which must be carefully balanced. The more information of a document is encoded in the fingerprint, the more reliably a possible collision of two fingerprints can be interpreted.
3. *Substring Size.* The substring size defines the fingerprint granularity. A fine granularity makes a fingerprint more susceptible to false matches, while with a coarse granularity fingerprinting becomes very sensitive to changes.

[1] The term "signature" is sometimes also used in this connection.

4. *Substring Encoding.* The selected substrings are mapped onto integer numbers. Substring conversion establishes a hash operation where—aside from uniqueness and uniformity—also efficiency is an important issue (Ramakrishna and Zobel (1997)). For this, the popular MD5 hashing algorithm is often employed (Rivest (1992)).

If the main issue is similarity analysis and not unique identification, the entire document d is used during the substring formation step—i. e., the union of all chunks covers the entire document. The total set of integer numbers represents the fingerprint $h(d)$. Note that the chunks may not be of uniform length but should be formed with the analysis task in mind.

2.1 Local and Global Similarity Analysis

For two documents A and B let $h(A)$ and $h(B)$ be their fingerprints with the respective resolutions $|h(A)|$ and $|h(B)|$. Following Finkel et al. (2002), a similarity analysis between A and B that is based on $h(A)$ and $h(B)$ measures the portion of the fingerprint intersection:

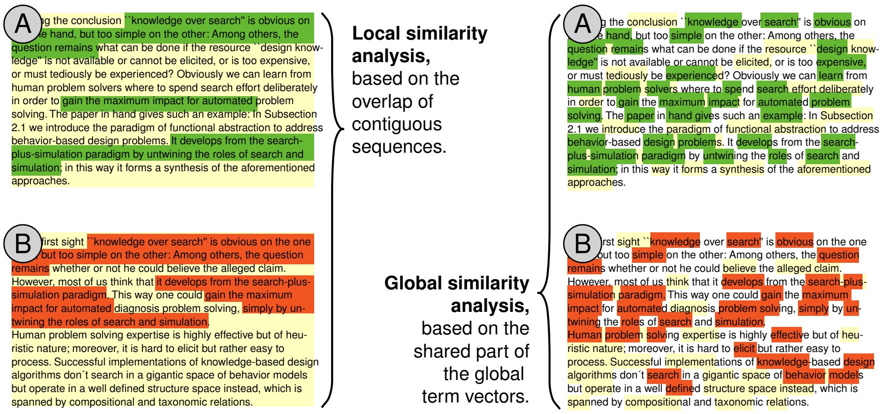
$$\varphi_{local}(A, B) = \frac{|h(A) \cap h(B)|}{|h(A) \cup h(B)|}$$

We call such a kind of similarity measure *local similarity* or *overlap similarity*, because it directly relates to the number of identical regions. By contrast, the vector space model along with the cosine measure does not depend on identical regions: Two documents may have a similarity of 1 though they may not share any 2-gram. The vector space model along with the cosine measure receives a global characteristic because it quantifies the term frequency of the entire document; in particular, the model neglects word order. Figure 1 contrasts the principles of local and global similarity analysis pictorially.

Basically, a fingerprint $h(d)$ of a document d is nothing more than a special document model of d. In this sense, every information retrieval task that is based on a standard document model can also be operationalized with fingerprints. However, fingerprint methods are more flexible since they can be targeted specifically towards one of the following objectives:

1. compactness—with respected to the document length
2. fidelity—with respected to a local similarity analysis

It is difficult to argue whether a fingerprint should be preferred to a standard document model in order to tackle a given information retrieval task. To better understand this problem of choosing an adequate document model we have developed a taxonomy of approaches to plagiarism analysis, which is shown in Figure 2. The approaches as well as the methods can be divided into local and global strategies. Note that in the literature on the subject local plagiarism analysis methods are encountered more often than global analysis methods. This is in the nature of things, since expropriating


A
B
Local similarity analysis,
based on the overlap of contiguous sequences.
Global similarity analysis,
based on the shared part of the global term vectors.
A
B

g the conclusion ``knowledge over search'' is obvious on e hand, but too simple on the other: Among others, the question remains what can be done if the resource ``design knowledge'' is not available or cannot be elicited, or is too expensive, or must tediously be experienced? Obviously we can learn from human problem solvers where to spend search effort deliberately in order to gain the maximum impact for automated problem solving. The paper in hand gives such an example: In Subsection 2.1 we introduce the paradigm of functional abstraction to address behavior-based design problems. It develops from the search-plus-simulation paradigm by untwining the roles of search and simulation; in this way it forms a synthesis of the aforementioned approaches.

first sight ``knowledge over search'' is obvious on the one , but too simple on the other: Among others, the question remains whether or not he could believe the alleged claim. However, most of us think that it develops from the search-plus-simulation paradigm. This way one could gain the maximum impact for automated diagnosis problem solving, simply by untwining the roles of search and simulation.
Human problem solving expertise is highly effective but of heuristic nature; moreover, it is hard to elicit but rather easy to process. Successful implementations of knowledge-based design algorithms don´t search in a gigantic space of behavior models but operate in a well defined structure space instead, which is spanned by compositional and taxonomic relations.

g the conclusion ``knowledge over search'' is obvious on e hand, but too simple on the other: Among others, the question remains what can be done if the resource ``design knowledge'' is not available or cannot be elicited, or is too expensive, or must tediously be experienced? Obviously we can learn from human problem solvers where to spend search effort deliberately in order to gain the maximum impact for automated problem solving. The paper in hand gives such an example: In Subsection 2.1 we introduce the paradigm of functional abstraction to address behavior-based design problems. It develops from the search-plus-simulation paradigm by untwining the roles of search and simulation; in this way it forms a synthesis of the aforementioned approaches.

rst sight ``knowledge over search'' is obvious on the one ut too simple on the other: Among others, the question remains whether or not he could believe the alleged claim. However, most of us think that it develops from the search-plus-simulation paradigm. This way one could gain the maximum impact for automated diagnosis problem solving, simply by untwining the roles of search and simulation.
Human problem solving expertise is highly effective but of heuristic nature; moreover, it is hard to elicit but rather easy to process. Successful implementations of knowledge-based design algorithms don´t search in a gigantic space of behavior models but operate in a well defined structure space instead, which is spanned by compositional and taxonomic relations.

Fig. 1. Two documents A and B which are analyzed regarding their similarity. The left-hand side illustrates a measure of local similarity: All matching contiguous sequences (chunks) with a length ≥ 5 words are highlighted. The right-hand side illustrates a measure of global similarity: Here the common word stems (without stopwords) of document A and B are highlighted. Observe that both similarity analyses may lead to the same similarity assessment.

the exact wording of another author often relates to text *passages* rather than to the entire text. At the second level our taxonomy differentiates the local approaches with respect to the comparison rigor, and the global approaches with respect to statistical analysis versus style analysis.

Among the shown approaches, the chunk identity analysis—usually operationalized with the MD5 hashing algorithm—is the most popular approach to plagiarism analysis. Nevertheless, the method comes along with inherent disadvantages: (*i*) it is computationally expensive, and (*ii*) a small chunk size must be chosen (3-10 words), which has a negative impact to both retrieval and storage performance. Observe that all mentioned problems can be countered, if the chunk size is drastically increased. This, however, requires some kind of fingerprints that operationalize a “relaxed” comparison concept.

The following subsection adresses this problem. It introduces fuzzy-fingerprints, which are specifically tailored to text documents and which provide the desired feature: an efficient means for near similarity analysis.

2.2 Fingerprints that Capture Near Similarity

While most fingerprint approaches rely on the original document d, from which chunks are selected and given to a hashing algorithm, our approach is based on the vector space model representation of the chunks. Key idea is a comparison of the distribution of the index terms in each chunk regarding their expected term frequency classes.

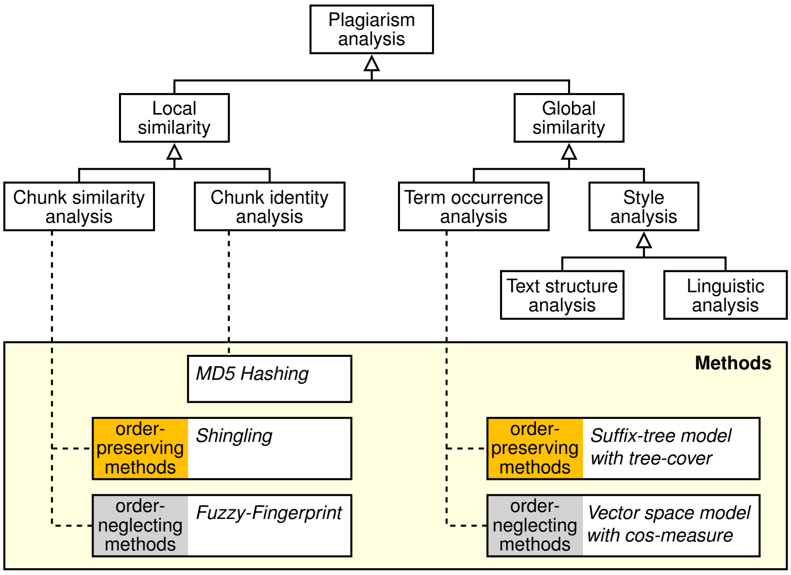


Fig. 2. A taxonomy of approaches and methods to plagiarism analysis.

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. Then, grounded on the analysis of large corpora a reference 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 chunk's term distribution from these a-priory probabilities forms a chunk-specific characteristic that can be encoded as small integer.

The basic procedure for constructing a fuzzy-fingerprint $h_\varphi(d)$ for a document d is as follows:[2]

1. Formation of a set $\mathcal{C}$ of chunks for d such that the extracted substrings $c \in \mathcal{C}$ cover d.
2. For each chunk $c \in \mathcal{C}$:
   (a) Computation of the vector space model $\mathbf{c}$ of c.
   (b) Computation of $\mathbf{pf}$, the vector of relative frequencies of the prefix classes for the index terms in $\mathbf{c}$.
   (c) Computation of Δ_{pf}, the vector of relative deviations of $\mathbf{pf}$ wrt. the expected prefix class distribution in the British National Corpus.

[2] Actually, the procedure is technically much more involved. It includes an alogrithm for chunking, the determination of suited prefix classes, the computation of a reference distribution, and the identification as well as application of fuzzification schemes. Details can be found in Stein (2005).

(d) Fuzzyfication of Δ_{pf} by abstracting the exact deviations towards a fuzzy deviation scheme with r intervals, and computation of a hash value γ:

$$\gamma = \sum_{i=0}^{k-1} \delta_i \cdot r^i, \qquad \text{with } \delta_i \in \{0, \ldots, r-1\}$$

k is the number of prefix classes, and δ_i is the fuzzified deviation of the frequency of prefix class i.

3. Formation of $h_\varphi(d)$ as the union of the hash values γ_c, $c \in \mathcal{C}$.

Remarks. The granularity of the fingerprint is controlled within three dimensions: By the number of chunks, $|\mathcal{C}|$, in Step 1, by the number of equivalence classes, k, in Step 2b, and by the resolution of the fuzzy deviation scheme, r, in Step 2d.

3 Runtime Performance and Classification Characteristic

This section presents results from a comparative analysis of the fuzzy-fingerprinting approach. In particular we investigate the following questions:

1. *Runtime Performance.* To which extent is plagiarism identification accelerated compared to MD5 fingerprinting?
2. *Classification Characteristic.* How does fuzzy-fingerprinting relate to other local (MD5 fingerprinting) and global (vector space model) similarity measures?

To answer these questions we set up different plagiarism experiments. The following plots result from a setting where as basis the RFC collection of the Internet Society was chosen: It comprises about 3000 documents with a considerable part of versioned sections. From this collection 50 documents were drawn randomly and compared to eight collection subsets with sizes between 100 and 800 documents. Since this comparison relied on the documents' fingerprint representations, the number of observed collisions corresponds directly to the runtime of the plagiarism analysis. Figure 3 reflects this fact: It shows the developing of the hash collisions (left) as well as the entire analysis time (right). The main reason for the large performance difference stems from the fact that fuzzy-fingerprinting allows for chunk sizes of 100 words on average, while MD5 fingerprinting works acceptable only for chunk sizes of 3 to 10 words.

The plot on the left-hand side in Figure 4 gives an answer to the question of how a document's local and global similarity analysis are related. It shows the deviation of fingerprint-based similarity values compared to the respective cosine similarity values under the vector space model, averaged over all documents of the RFC collection; observe that fuzzy-fingerprints resemble the cosine similarity better than MD5 fingerprints do. Especially against this background the plot on the right-hand side in Figure 4 must be interpreted:

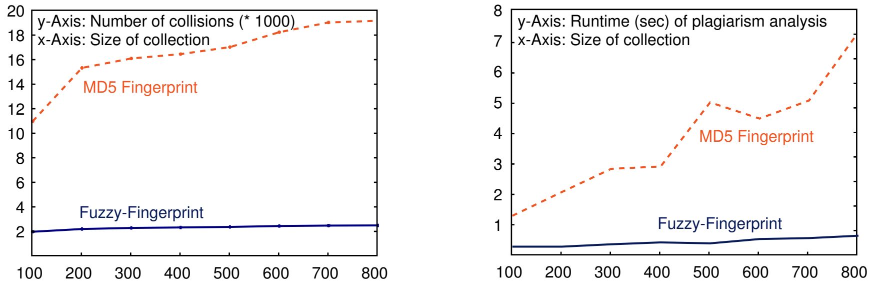


Fig. 3. Runtime performance of a plagiarism analysis task: 50 documents are compared to different subsets of the RFC collection. The figures show the runtime expressed in the number of fingerprint collisions (left) as well as in seconds (right).

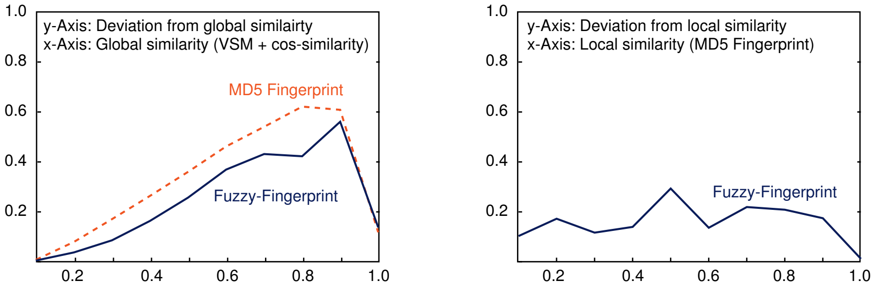


Fig. 4. Classification characteristics within the above plagiarism analysis task. Left: Similarity deviation of fingerprinting compared to the cosine similarity under the vector space model. Right: Similarity deviation of fuzzy-fingerprinting compared to optimum MD5 fingerprinting.

The rather small deviation between fuzzy-fingerprints and the "optimum" fingerprint, which is a fine-grained MD5 fingerprint of chunk size 3, illustrates the robustness of fuzzy-fingerprinting.

4 Summary

To identify plagiarized versions of a document or of some parts of it, similarity analyses must be performed. In this connection the paper introduced the distinction of local and global similarity measures. Local similarity measures answer the question which percentage of two documents are equal; global similarity measures answer the question to which percentage the entire documents are equal. This is a subtle but important difference, which leads to a taxonomy of methods for plagiarism analysis.

Local methods for plagiarism analysis base on fingerprinting, and in this paper we propose a new class of fuzzy-fingerprints that can be considered as an abstraction of the classical vector space model. These fingerprints allow for chunk sizes that are an order of magnitude larger than the typical MD5 digesting chunk sizes. As a consequence, the identification of plagiarism candidates is advanced significantly (more than a factor of five)—while reducing the size of the fingerprint database at the same time.

Our experiments also show the robustness of these fingerprints with respect to both large variations in the chunk size and the similarity range. Altogether, these properties make the concept of fuzzy-fingerprinting an ideal tool for plagiarism analysis and near similarity search in large document collections.

References

Brenda S. Baker. On finding duplication in strings and software. `http://cm.bell-labs.com/cm/cs/papers.html`, 1993.

Sergey Brin, James Davis, and Hector Garcia-Molina. Copy detection mechanisms for digital documents. In *SIGMOD '95*, pages 398–409, New York, NY, USA, 1995. ACM Press. ISBN 0-89791-731-6.

Encyclopædia Britannica. New Frontiers in Cheating. `http://www.britannica.com/eb/article?tocId=228894`, 2005.

Raphael A. Finkel, Arkady Zaslavsky, Krisztian Monostori, and Heinz Schmidt. Signature Extraction for Overlap Detection in Documents. In *Proc. of the 25th Australian conference on Computer Science*, pages 59–64. Australian Computer Society, 2002.

K. Fullam and J. Park. Improvements for scalable and accurate plagiarism detection in digital documents. `http://www.lips.utexas.edu/˜kfullam/pdf/DataMiningReport.pdf`, 2002.

Dan Gusfield. *Algorithms on Strings, Trees, and Sequences: Computer Science and Computational Biology*. Cambridge University Press, 1997.

Timothy C. Hoad and Justin Zobel. Methods for Identifying Versioned and Plagiarised Documents. *American Society for Information Science and Technology*, 54(3):203–215, 2003.

K. Monostori, R. Finkel, A. Zaslavsky, G. Hodsz, and M. Pataki. Comparison of overlap detection techniques. In *LNCS*, volume 2329, 2002.

Krisztián Monostori, Arkdy Zaslavsky, and Heinz Schmidt. Document overlap detection system for distributed digital libraries. In *DL '00*, pages 226–227, New York, NY, USA, 2000. ACM Press.

M. V. Ramakrishna and J. Zobel. Performance in Practice of String Hashing Functions. In *Proc. of the Intl. Conf. on Database Systems for Advanced Applications, Australia*, 1997.

Ronald L. Rivest. The md5 message-digest algorithm. `http://theory.lcs.mit.edu/r̃ivest/rfc1321.txt`, April 1992.

Narayanan Shivakumar and Hector Garcia-Molina. Building a scalable and accurate copy detection mechanism. In *DL '96*, pages 160–168, New York, NY, USA, 1996. ACM Press.

Antonio Si, Hong Va Leong, and Rynson W. H. Lau. Check: a document plagiarism detection system. In *SAC '97*, pages 70–77, New York, NY, USA, 1997. ACM Press.

Benno Stein. Fuzzy-Fingerprints for Text-based Information Retrieval. In Tochtermann and Maurer, editors, *5th Intl. Conf. on Knowledge Management (I-KNOW 05), Graz, Austria*, JUCS. Know-Center, 2005.