

# Near Similarity Search and Plagiarism Analysis

Benno Stein<sup>1</sup> and Sven Meyer zu Eissen<sup>2</sup>

<sup>1</sup> Faculty of Media, Media Systems

Bauhaus University Weimar, 99421 Weimar, Germany

<sup>2</sup> 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 these 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.<sup>1</sup> 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.

---

<sup>1</sup> 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:

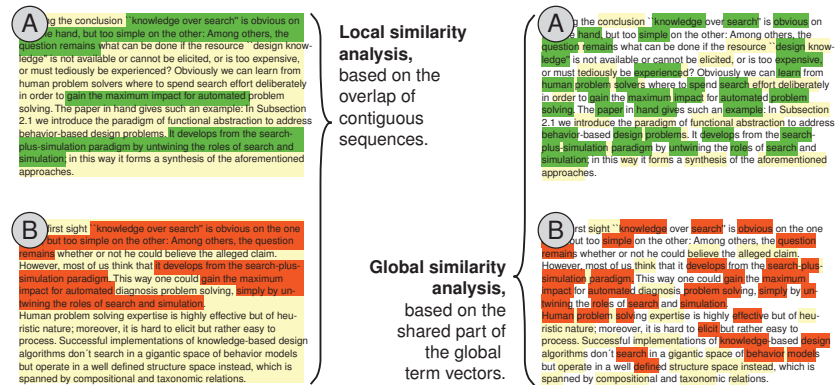
$$\varphi_{local}(A, B) = \frac{|h(A) \cap h(B)|}{|h(A) \cup h(B)|}$$

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



**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  $\geq 5$  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 addresses 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.

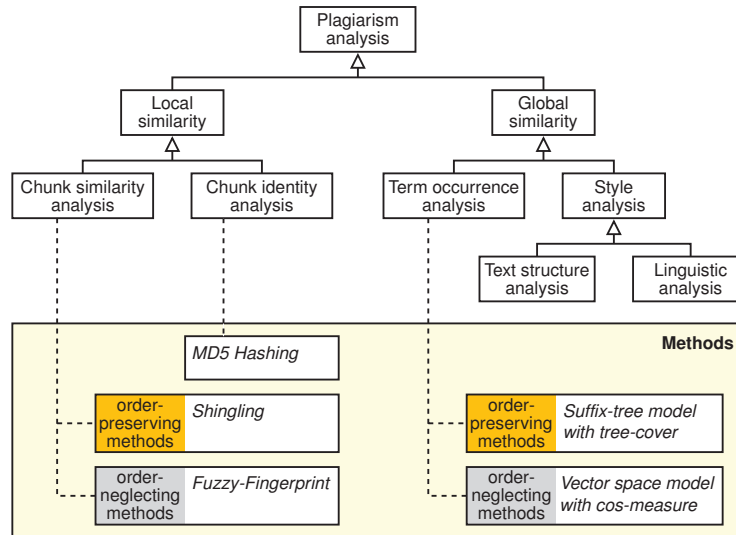


Fig. 2. A taxonomy of approaches and methods to plagiarism analysis.

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-priori 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-priori 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:<sup>2</sup>

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  $\Delta_{pf}$ , the vector of relative deviations of  $\mathbf{pf}$  wrt. the expected prefix class distribution in the British National Corpus.

<sup>2</sup> Actually, the procedure is technically much more involved. It includes an algorithm 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).

# Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

## Real-Time Litigation Alerts



Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

## Advanced Docket Research



With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

## Analytics At Your Fingertips



Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

## API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

## LAW FIRMS

Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

## FINANCIAL INSTITUTIONS

Litigation and bankruptcy checks for companies and debtors.

## E-DISCOVERY AND LEGAL VENDORS

Sync your system to PACER to automate legal marketing.