On Web-based Plagiarism Analysis

Alexander Kleppe, Dennis Braunsdorf, Christoph Loessnitz, Sven Meyer zu Eissen
Alexander.Kleppe@medien.uni-weimar.de,
Dennis.Braunsdorf@medien.uni-weimar.de,
Christoph.Loessnitz@medien.uni-weimar.de,
Sven.Meyer-zu-Eissen@medien.uni-weimar.de

Bauhaus University Weimar
Faculty of Media
Media Systems
D-99421 Weimar, Germany

Abstract The paper in hand presents a Web-based application for the analysis of text documents with respect to plagiarism. Aside from reporting experiences with standard algorithms, a new method for plagiarism analysis is introduced. Since well-known algorithms for plagiarism detection assume the existence of a candidate document collection against which a suspicious document can be compared, they are unsuited to spot potentially copied passages using only the input document. This kind of plagiarism remains undetected e.g. when paragraphs are copied from sources that are not available electronically. Our method is able to detect a change in writing style, and consequently to identify suspicious passages within a single document. Apart from contributing to solve the outlined problem, the presented method can also be used to focus a search for potentially original documents.
Key words: plagiarism analysis, style analysis, focused search, chunking, Kullback-Leibler divergence

1 Introduction

Plagiarism refers to the use of another's ideas, information, language, or writing, when done without proper acknowledgment of the original source [15]. Recently, the growing amount of digitally available documents contributes to the possibility to easily find and (partially) copy text documents given a specific topic: According to McCabe's plagiarism study on 18,000 students, about 50% of the students admit to plagiarize from Internet documents [7].

1.1 Plagiarism Forms

Plagiarism happens in several forms. Heintze distinguishes between the following textual relationships between documents: identical copy, edited copy, reorganized document, revisioned document, condensed/expanded document, documents that include portions of other documents. Moreover, unauthorized (partial) translations and documents that copy the structure of other documents can also be seen as plagiarized. Figure 1 depicts a taxonomy of plagiarism forms. Orthogonal to plagiarism forms are the underlying media: plagiarism may happen in articles, books or computer programs.

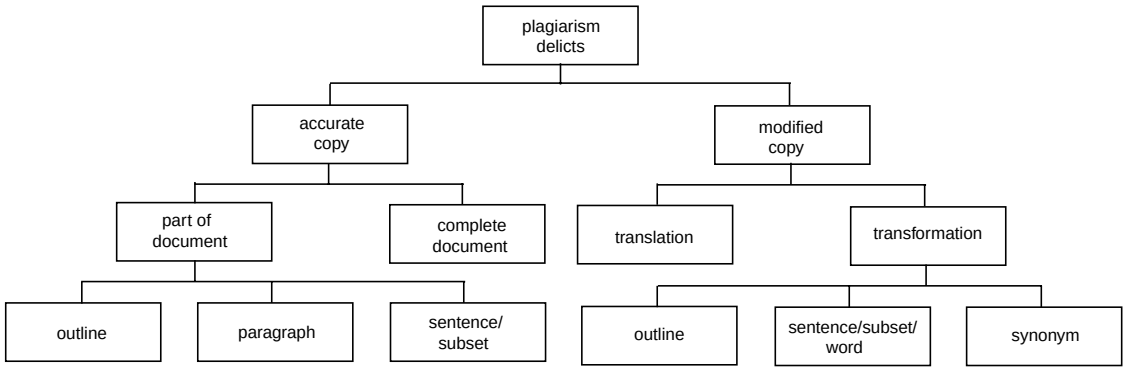

Figure 1. Taxonomy of plagiarism forms.

1.2 Plagiarism Analysis Process

Several challenges exist to find original sources for plagiarized documents. If no collection is given against which a suspicious document can be compared, it is reasonable to search for original sources on the Internet. When search engines like Google or Lycos are employed, the question is which keywords from the suspicious document deliver the most promising search results. Supposed that a keyword extraction algorithm is given, queries have to be generated that combine extracted keywords with respect to a selection strategy. The search results of the generated queries form a candidate document base. All documents from the candidate document base are represented through a document model that serves as abstraction for the analysis with one or more plagiarism detection algorithms. Figure 2 illustrates the process.

1.3 Related Work

Several methods for plagiarism analysis have been proposed in the past. Known methods divide a suspicious document as well as documents from the candidate base into chunks and apply a culling strategy to discard undesired chunks, e.g. too long or too short chunks. A hash function computes digital fingerprints for each chunk, which are inserted into a hash table: A collision of hash codes within the hash table indicates matching chunks.

Heintze's Koala system uses fingerprinting on fixed-length chunks [4]. Brin et al. experiments with sentence-based and hashed breakpoint chunking. Moreover, their discussion includes overlapping and non-overlapping chunking strategies [1]. Shivakumar and Garcia-Molina reports on performance of the aforementioned chunking strategies compared to word-based chunking. Monostori et al. proposes a suffix tree based post-processing method to filter out false positives. Finkel et al. introduce variance-based culling of chunks and discuss the use of text compression algorithms for plagiarism identification. In his PhD thesis, Monostori investigates parallel algorithms for plagiarism analysis. However, all of these approaches assume the existence of a document base against which a suspicious document can be compared.

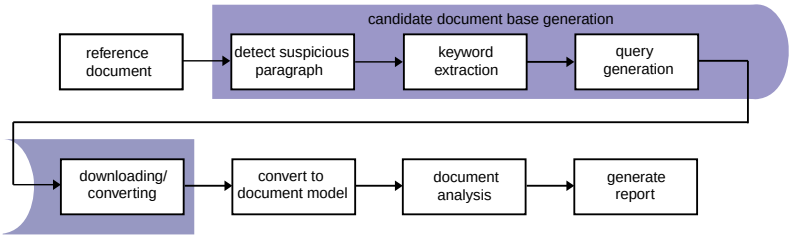

Figure 2. The plagiarism analysis process.

2 Generating a Candidate Document Base

If no document base is given, a candidate document base has to be constructed using search interfaces for sources like the Web, digital libraries, homework archives, etc. Standard search interfaces are topic-driven, i.e. they require the specification of keywords to deliver documents. Here keyword extraction algorithms come into play since extracted keywords serve as input for a query generation algorithm.

2.1 Keyword Extraction

Most of the keyword extraction algorithms are designed to automatically annotate documents with characteristic terms. Extracted keywords shall not only summarize and categorize documents, but also discriminate them from other documents. To automatically identify such terms is an ambitious goal, and several approaches have been developed, where each of which fulfills one or more of these demands. Existing keyword extraction algorithms can be classified by the following properties:

- *Supervision.* A keyword extraction algorithm that needs training data is called supervised; the remaining are called unsupervised.
- *Domain specialization.* Special purpose keyword extraction algorithms that are designed for a specific domain are called domain-specific. Often, domain-specific keyword extraction algorithms are supervised.
- *Document base.* The most part of the existing keyword extraction algorithms rely on a given document collection. Clearly, these algorithms perform better with respect to the discrimination power of identified keywords, since collection-specific measures like the inverse document frequency (idf) can be employed. However, extracting keywords from a single document is more challenging since respective algorithms must manage with less data.
- *Language specialization.* Some keyword extraction algorithms require language-dependent tools like term frequency normalization according to the distribution of terms within a language. These algorithms can be adapted to other languages as far as these tools are available in the target language.
- *Use of external knowledge.* (vs. purely statistical approaches).

Our Web-based plagiarism analysis application takes a suspicious document from an a-priori unknown domain as input. Consequently, an unsupervised, domain-independent keyword extraction algorithm that takes a single document as input would be convenient, language independence being a plus. Matsuo and Ishizuka propose such a method; it is based on a χ^2-analysis of term co-occurence data [6].

2.2 Query Generation: Focussing Search

When keywords are extracted from the suspicious document, we employ a heuristic query generation procedure, which was first presented in [12]. Let K_1 denote the set of keywords that have been extracted from a suspicious document. By adding synonyms, coordinate terms, and derivationally related forms, the set K_1 is extended towards a set K_2 [2]. Within K_2 groups of words are identified by exploiting statistical knowledge about significant left and right neighbors, as well as adequate co-occurring words, yielding the set K_3 [13]. Then, a sequence of queries is generated (and passed to search engines).

This selection step is controlled by *quantitative* relevance feedback: Depending on the number of found documents more or less "esoteric" queries are generated. Note that such a control can be realized by a heuristic ordering of the set K_3, which considers word group sizes and word frequency classes [14]. The result of this step is a candidate document collection $\mathcal{C} = \{d_1, \ldots, d_n\}$.

3 Plagiarism Analysis

As outlined above, a document may be plagiarized in different forms. Consequently, several indications exist to suspect a document of plagiarism. An adoption of indications that are given in [9] is as follows.

(1) *Copied text.* If text stems from a source that is known and it is not cited properly then this is an obvious case of plagiarism.
(2) *Bibliography.* If the references in documents overlap significantly, the bibliography and other parts may be copied. A changing citing style may be a sign for plagiarism.
(3) *Change in writing style.* A suspect change in the author's style may appear paragraph- or section-wise, e.g. between objective and subjective style, nominal- and verbal style, brillant and baffling passages.
(4) *Change in formatting.* In copy-and-paste plagiarism cases the formatting of the original document is inherited to pasted paragraphs, especially when content is copied from browsers to text processing programs.
(5) *Textual patchwork.* If the line of argumentation throughout a document is consequently incoherent then the document may be a "mixed plagiate", i.e. a compilation of different sources.

(6) *Spelling errors.* A change in the spelling of complicated technical terms is an interesting indication—which can directly be exploited to search for sources: Misspelled words and technical terms are salient keywords to use with search engines.
(7) *Outmoded diction.* Notably conspicious is the fact if text is copied from old books.

These indications are relatively easy to detect for an experienced human writer; the challenge is to find algorithms that operationalize the detection of these indications. Point (1) and partly point (2) require the analysis of external text sources; for this reason we call the associated analysis methods *relative*. The remaining indications can be examined without consulting foreign text and can be considered as *absolute*. The remainder of this section outlines existing work, which by now focuses on relative analysis only, and we introduce an absolute criterion in order to find changes in writing style.

3.1 Relative Plagiarism Analysis

Relative plagiarism analysis denotes the identification of plagiarized parts in a suspicious document d_0 with respect to a candidate document collection $\mathcal{C} = \{d_1, \ldots, d_n\}$. Let a document d_i be a sequence of $m(i)$ consecutive terms over a set of possible terms T, say $d_i = \{t_{i_1}, \ldots, t_{i_{m(i)}}\}$ with $t_{i_j} \in T$. A pair-wise comparison of term subsequences (chunks) of d_0 and d_i yields to the complexity $O(m(0)/k \cdot m(i)/k)$, given that a subsequence consists of k consecutive terms and a comparison is seen as an atomic unit, say, a single comparison takes $O(1)$. Assumed that d_0 and the documents in $\mathcal{C}$ have roughly the same length, then $m(i)/k = c$, with c being a constant. A comparison of d_0 with respect to the whole collection results in $O(\sum_{i=0}^{n} c^2) = O(nc^2)$. Observe that k will be relatively small compared to the document length since plagiarized parts to be found may be on the sentence level. Consequently, c will be a big constant and hence will influence the runtime performance significantly.

For this reason, past research has focused on comparing chunk fingerprints that are computed by a hash function $h : \mathcal{P}(T) \rightarrow \mathbf{N}$. The fingerprints $h(s)$ of chunks $s \subseteq d_0$ are inserted into a hash table; if a hash collision with chunks from documents in $\mathcal{C}$ happens, then the chunks are equal. This procedure reduces the complexity to $O(nc)$, assumed that h seeds well. A crucial point for this comparing method is the selection of an adeqate chunking strategy; known strategies fall in one of the following categories.

(1) *Fixed-size chunking.* Documents are split up into consecutive sequences of k terms (see above).
(2) *Sliding window chunking.* If term sequences of length k may overlap one speaks of sliding window chunking.
(3) *Sentence chunking.* Here a sequence of terms is terminated by a sentence delimiter, e.g. ".", "?" or "!". Problems arise when abbreviations like “e.g.” are used

within a text. Note that abbreviations are not a closed-class word set since abbreviations may develop and evolve over time, take for example "U.S." or "w.r.t." [1]. Other critical elements comprise fomulas or Internet addresses.

(4) *Hashed breakpoint chunking.* A sequence of terms in d_i is terminated at position j if the equation $h(t_{i_j}) \, mod \, z = 0$ holds, where $z \in \mathbf{N}$ is fixed and chosen a-priori. z controls the expected text chunk length if h's values are equally distributed. A property of hashed breakpoint chunking is that all terms with $h(t_{i_j}) \, mod \, z = 0$ act chunk-synchronizing between documents. This property becomes important when copied text is edited.

Another research direction appeared recently. Cryptographic hash functions that were used to fingerprint chunks in the past come along with the property $h(s_1) = h(s_2) \Rightarrow s_1 = s_2$ with high probability for any pair of chunks s_1, s_2. Assumed that plagiarized text is slightly edited, it is desirable to find fuzzy hash functions h_F that relax this condition to $h_F(s_1) = h_F(s_2) \Rightarrow sim(s_1, s_2) \geq 1 - \varepsilon$, with sim denoting a text similarity function. Research that addresses plagiarism detection in this connection can be found in [11].

3.2 Absolute Analysis

As pointed out at the beginning of this section, a variety of plagiarism indications exist that can be identified without comparing text against a collection of candidate documents. In the following we propose a method to identify writing style changes as prescribed in the indications, under point (3).

Style cannot be measured directly, but the frequency of used word classes can characterize an author's style. We employ a part-of-speech analysis to measure the global distribution of word classes in a document (cf. Table 3.2). The result is a set of 18 word class attributes that encode the relative frequency of word classes. Assumed that a paragraph in the document is copied from another author, the writing style in this paragraph may be different. We compare the local word class distribution in each paragraph to the global distribution. If a local distribution diverges significantly from the global distribution, the associated paragraph may be copied. The divergence can be measured using the Kullback-Leibler criterion [5]. Let W denote the set of word classes, and let $p(w)$ ($q(w)$) denote the local (global) relative word class frequency for each $w \in W$. The Kullback-Leibler divergence measure is defined as

$$KL_W(p, q) = \sum_{w \in W} p(w) \log \frac{p(w)}{q(w)}$$

Note that this criterion applies for plagiarism cases where some paragraphs are copied into a longer document. It may also help to identify "patchworked texts" in which the variance of divergences is expected to be high.

Attribute	Target	Attribute	Target
w_1	adjective	w_{10}	interjection
w_2	adverb	w_{11}	modal
w_3	alphanumeric term	w_{12}	noun
w_4	article	w_{13}	preposition
w_5	the word “be”	w_{14}	pronoun
w_6	copula	w_{15}	relative pronoun
w_7	the word “do”	w_{16}	symbol
w_8	foreign word	w_{17}	the word “to”
w_9	the word “have”	w_{18}	verb

Table 1. Attribute set W for style analysis. The attributes measure the relative frequency of the denoted target word class.

4 Experiences

As already pointed out, no plagiarism analysis test collection is available. However, interesting questions to be answered experimentally with respect to KL_W include the following:

(1) With which precision/recall does KL_W detect a foreign paragraph that was copied from one document to another?
(2) Up to which amount of copied text in a document does KL_W work reliably?
(3) Is KL_W genre-dependent?
(4) Can KL_W detect patchworked documents?
(5) How many false positives are delivered by KL_W when applied to original documents?
(6) How does the editing of copied text influence the performance of KL_W?
(7) How does an a-priori identification of suspicious paragraphs influence the quality of search results when keywords are extracted from suspicious paragraphs only?

We compiled a test collection of 40 documents from the Internet that we used to generate 360 plagiates to test the style ananlysis criterion. One finding that relates to question (1) is that KL_W detects about 80% of copied paragraphs, given that a copied paragraph comprises more than 50 terms and the reference document has a length of at least 5 kb. We are currently building a large plagiarism test collection that lets us answer the remaining questions on a statistically founded basis; however, the design of this collection goes beyond the scope of this paper.

The query generation algorithm that was given in Subsection 2.2 delivered all known original texts when downloading 50 documents. Another unexpected finding is that one text in our test collection emerged as partly plagiarized from Wikipedia. This fact shows that in contrast to fixed collections it is hard to evaluate Web-based tools for plagiarism analysis: Not all potential sources for plagiarism can be listed in a collection since the Web evolves, and copies may appear or disappear every now and then.

Figure 3 shows a screenshot of our prototype. A suspicious document that was uploaded to be analyzed by our prototype is depicted on the left; on the right a document that has been found on the Web using the techniques described above is shown. Passages that match are marked in red. Each bar in the chart on the left shows a KL_W value. Other matching documents from the Web can be selected to view using the Tabs (top right).

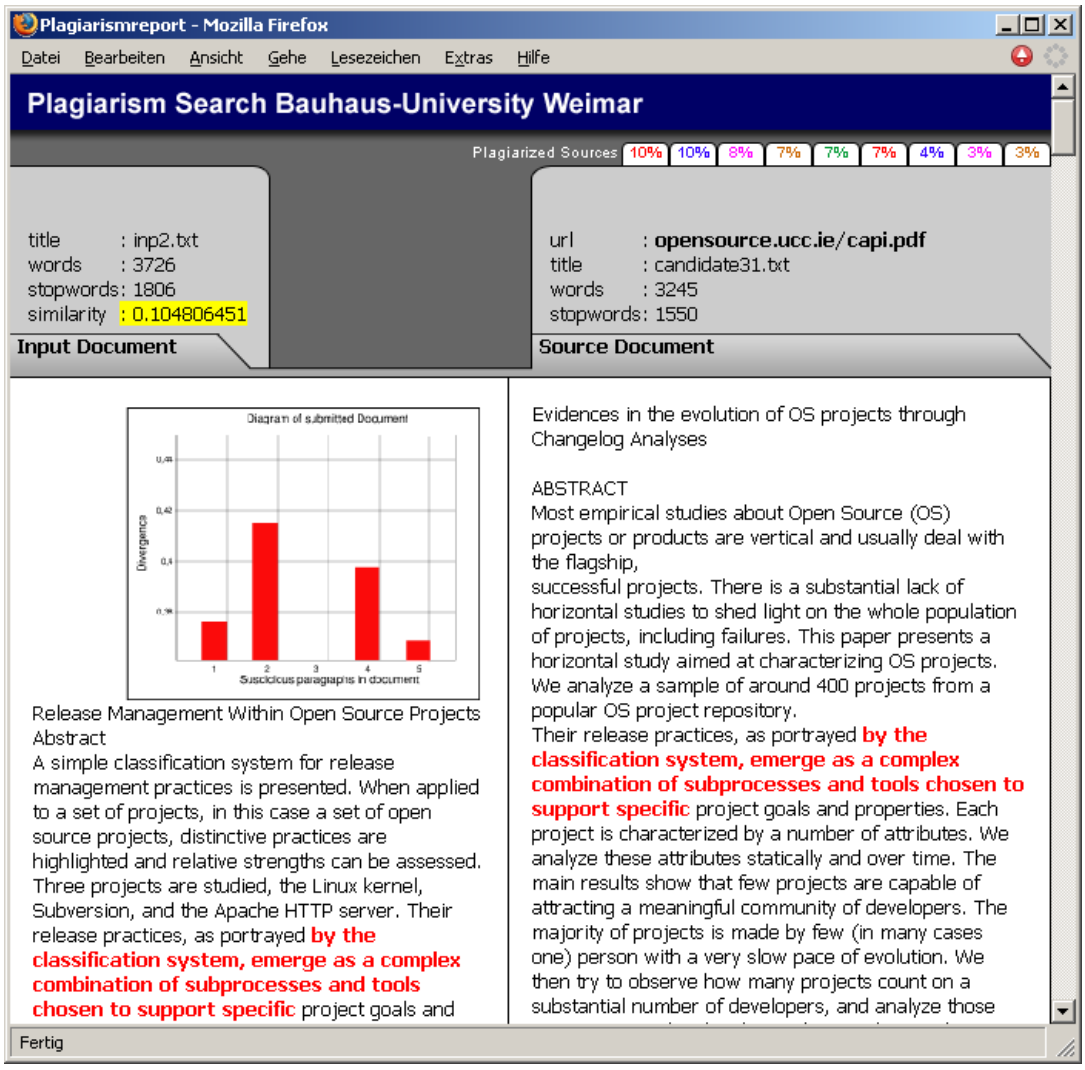

Figure 3. Screenshot of our prototype. The red text marks matching passages of a suspicious document (left) and a document that has been found on the internet (right). Each bar in the chart on the left shows a KL_W value. Other matching documents from the Web can be selected to view using the Tabs (top right).

5 Conclusion

In this paper, we showed how to build a Web-based application for the analysis of potentially plagiarised documents on the Internet using state-of-the art technology. We distinguished between absolute and relative plagiarism analysis and introduced a new (absolute) writing style analysis method.

In future, algorithms for the automatic detection of plagiarism indications that have been listed in Section 3 have to be designed; moreover, the experiences have shown that effective evaluation methodology for Web-based plagiarism analysis has to be developed. The next version of our prototype we will substitute cryptographic fingerprints with fuzzy-fingerprints.

Bibliography

[1] S. Brin, J. Davis, and H. 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.

[2] C. Fellbaum. *WordNet: An Electronic Lexical Database*. MIT Press, 1998.

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

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

[5] S. Kullback and R. A. Leibler. On information and sufficiency. *Annals of Mathematical Statistics*, 22(1):79–86, mar 1951.

[6] Y. Matsuo and M. Ishizuka. Keyword extraction from a single document using word co-ocurrence statistical information. *Int'l Journal on Artificial Intelligence Tools*, 13(1):157–169, 2004.

[7] D. McCabe. Research Report of the Center for Academic Integrity. `http://www.academicintegrity.org`, 2005.

[8] K. Monostori, A. Zaslavsky, and H. Schmidt. Efficiency of data structures for detecting overlaps in digital documents. In *ACSC '01: Proceedings of the 24th Australasian conference on Computer science*, pages 140–147, Washington, DC, USA, 2001. IEEE Computer Society. ISBN 0-7695-0963-0.

[9] F. Schätzlein. Studentischer Trendsport 'copy-and-paste': Plagiate erkennen, überprüfen und verhindern. *ZWO. E-Journal des Instituts für Germanistik II der Universität Hamburg*, 2, 2003.

[10] N. Shivakumar and H. Garcia-Molina. Building a scalable and accurate copy detection mechanism. In *DL '96*, pages 160–168, New York, NY, USA, 1996. ACM Press. ISBN 0-89791-830-4.

[11] B. Stein. Fuzzy-Fingerprints for Text-based Information Retrieval. In Tochtermann and Maurer, editors, *5th International Conference on Knowledge Management (I-KNOW 05), Graz, Austria*, JUCS. Know-Center, 2005.

[12] B. Stein and S. Meyer zu Eissen. Automatic market forecast summarization from internet data. In *Proc. of the IADIS WWW/Internet Conference*, 2005.

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

[14] E. M. Voorhees. Query expansion using lexical-semantic relations. In *SIGIR '94*, pages 61–69, New York, NY, USA, 1994. Springer-Verlag New York, Inc. ISBN 0-387-19889-X.

[15] Wikipedia. Plagiarism. `http://en.wikipedia.org/wiki/Plagiarism`, 2005.