2008-12-09

Nowhere to Hide: Finding Plagiarized Documents Based on Sentence Similarity

Nathaniel Gustafson

Yiu-Kai D. Ng
ng@cs.byu.edu

See next page for additional authors

Follow this and additional works at: https://scholarsarchive.byu.edu/facpub

Part of the Computer Sciences Commons

Original Publication Citation

BYU ScholarsArchive Citation
Gustafson, Nathaniel; Ng, Yiu-Kai D.; and Pera, Maria Soledad, "Nowhere to Hide: Finding Plagiarized Documents Based on Sentence Similarity" (2008). All Faculty Publications. 150.
https://scholarsarchive.byu.edu/facpub/150

This Peer-Reviewed Article is brought to you for free and open access by BYU ScholarsArchive. It has been accepted for inclusion in All Faculty Publications by an authorized administrator of BYU ScholarsArchive. For more information, please contact scholarsarchive@byu.edu, ellen amatangelo@byu.edu.
Authors
Nathaniel Gustafson, Yiu-Kai D. Ng, and Maria Soledad Pera

This peer-reviewed article is available at BYU ScholarsArchive: https://scholarsarchive.byu.edu/facpub/150
Nowhere to Hide: Finding Plagiarized Documents Based on Sentence Similarity

Nathaniel Gustafson  Maria Soledad Pera  Yiu-Kai Ng
Computer Science Department  Brigham Young University  Provo, Utah, U.S.A.

Abstract

Plagiarism is a serious problem that infringes copyrighted documents/materials, which is an unethical practice and decreases the economic incentive received by authors (owners) of the original copies. Unfortunately, plagiarism is getting worse due to the increasing number of on-line publications on the Web, which facilitates locating and paraphrasing information. In solving this problem, we propose a novel plagiarism-detection method, called SimPaD, which (i) establishes the degree of resemblance between any two documents \( D_1 \) and \( D_2 \) based on their sentence-to-sentence similarity computed by using pre-defined word-correlation factors, and (ii) generates a graphical view of sentences that are similar (or the same) in \( D_1 \) and \( D_2 \). Experimental results verify that SimPaD is highly accurate in detecting (non-)plagiarized documents and outperforms existing plagiarism-detection approaches.

1 Introduction

Plagiarism, which is a prolific problem, especially in the academic world, is getting worse, since the volume of on-line publications has been increasing during the past decades. Common plagiarism methods either simply duplicate material from a (non-)electronic source, or copy material from a given source and intentionally modify its wordings or sentence structures without affecting its content [7]. The latter is more difficult to identify due to its complexity.

Popular plagiarism-detection approaches (i) compute the overlapping among n-grams in any two documents [9], (ii) analyze the writing, i.e., syntactical and grammatical, styles of the authors of various documents [15], (iii) identify words substituted by their synonyms and split/merged sentences [16], and (iv) detect plagiarized documents based on their fingerprints [8]. Besides using synonyms, hypernyms, and hyponyms, majority of these approaches rely on exact word/phrase matching in finding the portion of a document that is plagiarized, which unfortunately is insufficient and inaccurate, since it is a common practice to paraphrase words by using similar ones for plagiarizing a source document. In this paper, we propose a new, novel plagiarism-detection approach which considers not only (similar) word substitution, addition, and deletion, but also sentence splitting and merging based on word-similarity measures.

The proposed plagiarism-detection method, called Similarity-based Plagiarism Detection (SimPaD) approach, conducts sentence-to-sentence comparison. For any two given documents \( D_1 \) and \( D_2 \), SimPaD determines the degree of resemblance between \( D_1 \) and \( D_2 \) using the pre-computed word-correlation factors defined in [10], which can be applied for detecting exact and semantically-related, words in different sentences to determine the degree of resemblance between any two (words/sentences in) given documents, a simple and computational effective process. SimPaD can detect plagiarized documents by identifying (i) sentences in a plagiarized document that are split/merged from sentences in a source document as well as (ii) sentences in a plagiarized document in which words have been deleted from, added to, or replaced by others in the original sentences of a source document but retain similar content. Unlike existing plagiarism-detection approaches, SimPaD is unique, since it (i) allows partial similarity matching as opposed to the strict exact matches, and (ii) uses a graphical view to display the plagiarized sentences in a plagiarized document matched with the corresponding sentences in a source document (based on their degrees of similarities). Experimental results show that SimPaD is highly accurate in detecting (non-)plagiarized documents.

This paper is organized as follows. In Section 2, we discuss existing plagiarism-detection methods. In Section 3, we present SimPaD. In Section 4, we evaluate the performance of SimPaD. In Section 5, we give a conclusion.

2 Related Work

Many attempts have been made in the past to detect plagiarized documents. In [8], Lukashenko et al. compare
two documents and determine their degree of similarity using different metrics such as the Euclidean distance, Cosine similarity, the percentage of shared n-grams, and the resemblance among estimated language models, whereas Monostori et al. [9] present a plagiarism-detection system, denoted MatchDetect Reveal (MDR), using suffix-trees. MDR, which is capable of detecting overlapping in (potential) plagiarized documents, applies a string-matching algorithm to identify suspicious documents from where suffix-trees are constructed using a modified Ukkonen’s algorithm.

In [7], the authors propose using a natural language processing method to facilitate the detection of plagiarized documents not only among the ones created by “cut and paste,” but also documents in which both the text and the structure of the original sentences are altered, while the content of the documents are not affected. This approach, which considers (i) word replacement with assigned weights to exact matches, synonyms, hypernyms, etc. when performing sentence-to-sentence comparisons to establish the degree of resemblance among sentences, and (ii) syntactic (semantic, respectively) processing to analyze the syntactic structure (meaning, respectively) of the sentences, works well only when the two documents to be compared are highly similar.

In [5], Khmelev et al. use the R-measure to detect plagiarized documents. The R-measure adds the lengths of the substrings in a given document that are included in another one in a collection. By considering the normalized R-measure value, it is possible to establish the “repeatedness” of a document with respect to others, which establishes the degree of plagiarism in the corresponding documents.

Tashiro et al. introduce EPCI [12], which is a tool for finding copyright infringement texts. Given a potential plagiarized document D, EPCI extracts several sequences of words, i.e., seed text, and generates queries using the seed text to retrieve a set of Web documents W that could be the source of the content of D. Hereafter, EPCI computes the similarity between D and the documents in W. The higher the similarity value between D and any document in W, the more likely that infringement has occurred.

3 Our Plagiarism Detection Approach

Plagiarism can be detected by establishing the “content similarity” among documents [15]. SimPaD identifies Dp as a plagiarized document from a source document Ds, if Dp contains (words in) sentences with high degrees of similarity to (words in) sentences of Ds. In reality, plagiarism detection is not as simple as matching sentences with sentences, since sentences in Ds may not be copied entirely into Dp, i.e., a “cut and paste” plagiarism; instead, they could be reordered, split, and merged, and/or have words in them added to, deleted from, or replaced in Dp. Indeed, establishing which sentences of Ds have been plagiarized is quite broad in scope. For this reason, SimPaD considers a number of integrated plagiarism-detection strategies on words and sentences, which are discussed in the following subsections. SimPaD applies these strategies in tandem, rather than independently, which complement each other in determining plagiarized sentences/documents.

3.1 Document Representation

Prior to analyzing potential plagiarized documents, we first remove all the stopwords and reduce all the non-stopwords in a document D to their grammatical roots, i.e., stems. In addition, as part of the pre-processing step, short sentences are removed from D due to the high probability that independent authors can create (semantically the) same short sentences rather than long, similar ones, which are less likely similar by chance.

Example 1 Consider the sentences in the following two small documents, D1 and D2:

D1: “Many people believe that lemmings are prone to frequently jumping off a cliff in mass suicide. This is not true.”

D2: “One may assume that this chemical reaction is unfeasible due to the steric hindrance. This is not true.”

Clearly, D1 and D2 are different in content, and neither one is plagiarized from another. However, the sentence “this is not true” appears in both documents, which is accounted to the tendency of some words and sentences that naturally appear more frequently regardless of authorship.

We exclude sentences from documents to be evaluated by SimPaD that are sufficiently short. In [3], Gildea estimates the average number of words in an English sentence varies between 15 and 20 words, whereas LaRocque [6] treats every sentence with less than 12 words (including stopwords) as short. Hence, we remove (short) sentences with fewer than 7 non-stop, stemmed words during the process of plagiarism detection.

3.2 Manipulation of Words

Words in a source sentence may have been reordered, substituted, deleted, or added to yield a plagiarized sentence. We compute the similarity values of words in sentences for detecting plagiarized sentences/documents.

\footnote{Stopwords are words that have little meaning, such as articles, conjunctions, and prepositions, which can be removed from a document without significant information loss. According to a study based on the TREC corpora [17], at least 30% of the words in a document are stopwords. Moreover, relevance rankings on documents excluding stopwords consistently outperform the ones on documents including stopwords [13].}
3.2.1 Word Reordering

It is quite common that a plagiarized sentence is created from a source sentence $S$ by reordering the words in $S$. In the simplest case of word reordering, the same keywords\(^2\) in $S$ are present and placed in a different order, along with probably additional words, in a plagiarized sentence $P$.

Example 2 Consider the following source sentence $S$ and plagiarized sentence $P$:

$S$: “Over 45% of all current high school students are involved in intramural sports of some kind.”

$P$: “Of all the current high school students, over 45% are involved in some kind of intramural sports.” □

As shown in Example 2, the order of words does not affect the content of $P$ and $S$.\(^3\) Thus, $SimPaD$ discards the order of words in comparing any (sentences in) documents.

3.2.2 Word Substitution

Word substitution can be viewed as deleting a word in a source sentence $S$ followed by adding a (similar) word in $S$.

Example 3 Consider the following sentences $S$ and $P$:

$S$: “Many dairy farmers today use machines for operations from milking to culturing cheese.”

$P$: “Today many cow farmers perform different tasks from milking to making cheese using automated devices.” □

As stated in [15], the problem of word substitution is a complex one to address in plagiarism detection, partially due to the lack of plagiarism-detection schemes which measure the degrees of similarity among words. In developing such a scheme for determining content similarity of (words in) any two sentences, we first consider how a human may compare words in them. Consider the sentences in Example 3. A person may initially notice several identical words in both $S$ and $P$, and further evaluating the content of each sentence shows that “making cheese” (“automated devices” and “tasks”, respectively) is quite similar to “culturing cheese” (“machines” and “operations”, respectively). A significant number of (non-)identical words with similar/same meaning in two sentences provide solid evidence that the sentences come from the same origin.

3.2.3 Word Addition/Deletion

A word deleted from (added to, respectively) a sentence without adding (deleting, respectively) another word can be considered as a special case of word substitution. We realize that the similarity of sentences $P$ and $S$ is higher when words added to $P$ are closely related to (or the same as) the words in $S$. However, adding non-related words (in terms of similarity with the words in $S$) to $P$ yields lower sentence-to-sentence similarity of $P$ with respect to $S$.

3.2.4 Word-Correlation Factors

In establishing the degrees of similarity among non-identical keywords for plagiarism detection, we adapt the word-correlation factors defined by [10] in a pre-computed word-correlation matrix. The word-correlation factors between any two words $i$ and $j$, denoted $Sim(i,j)$, were pre-computed using 880,000 documents in the Wikipedia collection (downloaded from http://www.wikipedia.org/\(^4\)) based on their (i) frequency of co-occurrence and (ii) relative distance in each Wikipedia document as defined below.

$$Sim(i,j) = \frac{\sum_{w_{i} \in V(i)} \sum_{w_{j} \in V(j)} \frac{1}{d(w_{i},w_{j})+1}}{|V(i)| \times |V(j)|}$$ (1)

where $d(w_{i}, w_{j})$ is the distance between any two words $w_{i}$ and $w_{j}$ in any Wikipedia document $D$, $V(i)$ ($V(j)$, respectively) is the set of stem variations of $i$ ($j$, respectively) in $D$, and $|V(i)| \times |V(j)|$ is the normalization factor.

The Wikipedia collection is an ideal and unbiased choice for establishing word similarity, since (i) documents within the collection were written by close to 90,000 authors with different writing styles and word usage, (ii) the Wikipedia documents cover an extensive range of topics, and (iii) the words within the documents appear in a number of on-line dictionaries, such as 12dicts-4.0, Ispell, and BigDict. Compared with the word-correlation factors, WordNet (http://wordnet.princeton.edu/) provides synonyms, hypernyms, holonyms, antonyms, etc. for a given word. There is, however, no partial degree of similarity measures, i.e., weights, assigned to any pair of words. For this reason, the word-correlation factors yield a more sophisticated measure of similarity of words than the words in WordNet.

3.2.5 N-gram Correlation Factors

As mentioned in Section 3.2.1, $SimPaD$ does not consider the order of words in sentences. However, in some cases, disregarding the order of the words in a sentence might yield a higher degree of similarity of sentences than necessary, which could falsely classify a legitimate document as plagiarized, generating a false positive. In order to reduce the number of false positives, we can consider n-gram, phrase-correlation factors ($2 \leq n \leq 3$), which are

\(^2\)From now on, (key)words refer to non-stop, stemmed words.

\(^3\)Word-reordering has been widely-used in modern plagiarism approaches. See [15] for details.

\(^4\)Words within the Wikipedia documents were stemmed (i.e., reduced to their root forms) and stopwords were removed.
computed by combining the correlation factors of the corresponding words in the n-grams of sentences to be compared as defined in [10], if needed. Since experimental results (presented in Section 4) show that the Sim values on words are adequate in detecting (non-)plagiarized documents accurately, n-gram phrase-correlation factors are not further considered for plagiarism detection in this paper.

3.3 Sentence Similarity

SimPaD computes the degree of similarity of any two sentences using

\[
\text{LimSim}(P, S) = \frac{\sum_{i=1}^{m} \min(1, \sum_{j=1}^{n} \text{Sim}(i, j))}{m}
\]

where \(m\) (\(n\), respectively) denotes the number of keywords in a (potential) plagiarized (source, respectively) sentence \(P\) (\(S\), respectively), \(i\) (\(j\), respectively) is a word in \(P\) (\(S\), respectively), and \(\text{Sim}(i, j)\) is the word-correlation factor of \(i\) and \(j\). 

LimSim\((P, S) \neq \text{LimSim}(S, P)\), unless \(P=S\).

Using the LimSim function, instead of simply adding the Sim value of each word in \(P\) with respect to each word in \(S\), we restrict the highest possible sentence-similarity value between \(P\) and \(S\) to 1, which is the value for exact matches. By imposing this constraint, we ensure that if \(P\) contains a word \(k\) that is (i) an exact match of a word in \(S\), and (ii) similar to (some of) the other words in \(S\), then the degree of similarity of \(P\) with respect to \(S\) cannot be significantly impacted/affected by \(k\) to ensure a balanced similarity measure of \(P\) with respect to \(S\).

3.3.1 Merged/Split Sentences

Besides considering word addition, deletion, and substitution in detecting plagiarism, we identify sentences in a plagiarized document \(D_P\) created by splitting and/or merging sentences in a source document \(D_S\). Identifying these split/merged sentences in \(D_P\) not only measures the document similarity of \(D_P\) with respect to \(D_S\) more accurately, this information is also useful to SimPaD users who are interested in knowing which sentences in \(D_S\) have been split/merged to yield the corresponding sentences in \(D_P\).

Some plagiarism-detection methods, such as [16], consider sentence rearrangement, i.e., sentence merging and splitting, by setting a threshold value \(V\) so that each pair of sentences with a number of words in common that is higher than \(V\) is further evaluated. Relying on the proportion of common words among sentences for detecting split/merged sentences, however, is a limitation, since as previously mentioned, words in a given source sentence \(S\) may have been replaced by other similar, but not the same, ones to yield a plagiarized sentence \(P\), and hence the number of common words between \(S\) and \(P\) is lower than what it should be.

We claim that a split sentence \(P\) is “subsumed” by its original sentence \(S\) if majority of the words in \(P\) are (semantically) the same as (some of) the words in \(S\). By adopting the threshold value of 0.93, which was established and verified using text documents in [10], SimPaD treats \(P\) as a split (subsumed) sentence from (of) \(S\) if \(\text{LimSim}(P, S) \geq 0.93\). The same strategy can be applied to detect merged sentences, i.e., source sentences \(S_1, \ldots, S_n\) (\(n \geq 2\)) are merged to yield a plagiarized sentence \(P\), if \(\text{LimSim}(S_i, P) \geq 0.93, 1 \leq i \leq n\).

3.3.2 Sentence-to-Document Similarity

Using the LimSim value of each sentence \(P\) in a (potential) plagiarized document \(D_P\) with respect to each sentence in a source document \(D_S\), SenSim\((P, D_S)\) can identify the highest degree of similarity of \(P\) with sentences in \(D_S\), which yields the probability of \(P\) having the same content as a sentence in \(D_S\), and is defined as

\[
\left\{ \begin{array}{ll}
\max(\forall S_j \in D_S \text{LimSim}(P, S_j)), & \text{if there exists at most one } S_j \text{ such that } \text{LimSim}(S_j, P) \geq 0.93 \\
\min(1, \sum_{j=1}^{n} \text{LimSim}(P, S_j)) & \text{such that } \text{LimSim}(S_j, P) \geq 0.93, \text{otherwise}
\end{array} \right.
\]

where \(n\) is the number of sentences in \(D_S\) that are subsumed by \(P\). SenSim\((P, D_S)\) returns the highest LimSim of \(P\) with respect to the sentences in \(D_S\), if \(P\) is not created by merging two or more sentences in \(D_S\); otherwise, the combined similarity of the sentences in \(D_S\) that are merged to yield \(P\) is assigned to be the sentence-to-document value of \(P\) with respect to \(D_S\). Using the Min value in Equation 3, we impose the same restriction as in Equation 2, i.e., we limit the combined LimSim values to 1, an exact match.

3.3.3 Dotplot Views of Similar Sentences

Using the SenSim (LimSim, respectively) values computed by Equation 3 (Equation 2, respectively), we can (i) identify for each sentence in a (potential) plagiarized document \(D_P\) its most highly-related sentence in a source document \(D_S\), in addition to sentences in \(D_P\) that are split/merged sentences from sentences in \(D_S\), and (ii) graphically display these related sentences.

Dotplot view [4] was designed for visualizing patterns of string matches in different kinds of texts, e.g., news articles, programming code, etc. We use the Dotplot view to provide an intuitive, conceptual diagram that shows similar sentences in a source and a plagiarized document visually. We did modify, however, the Dotplot view using the scatter graph in Microsoft Office Excel and call the modified graph Plagiarism View (or PlaView). In each PlaView, the \(x\)- (\(y\), respectively) axis represents the sentences (by numbers in a chronological order of their appearance).
in a plagiarized document \(D_P\) (source document \(D_S\), respectively), whereas each dot, denoted “•”, in PlaView represents the sentence \(S\) in \(D_S\) that is the most highly similar to the sentence \(P\) in \(D_P\), i.e., \(\text{SenSim}(P, D_S)\). Furthermore, PlaView graphically displays the sentences \(P_1, \ldots, P_n\) in \(D_P\) that are the split version of a sentence \(S\) in \(D_S\), if \(\text{LimSim}(P_i, D_S) \geq 0.93\), and the “dots” of \((P_1, S), \ldots, (P_n, S)\), \(1 \leq i \leq n\), are horizontally aligned in PlaView. In addition, a cross symbol, i.e., “×”, in PlaView denotes a merged sentence \(P\) in \(D_P\) that combines several sentences \(S_1, \ldots, S_m\) \((m \geq 2)\) in \(D_S\), such that \(\text{LimSim}(S_i, P) \geq 0.93, 1 \leq i \leq m\), and the “crosses” of \((S_i, P)\) are vertically aligned. Furthermore, the larger the dot (cross, respectively) size, the higher the content similarity of the corresponding sentences.

Figure 1 shows the PlaView of the document “Student File Management under Primos” and its plagiarized version in Webis-PC, which is one of the datasets used in Section 4, and \(v_1\) and \(v_2\) of \(<v_1, v_2>\) in Figure 1 denote the \(\text{SenSim}\) and \(\text{LimSim}\) values of \(P\) and \(D_S\), and \(P\) and \(S\), respectively.

### 3.4 Document Similarity

Having identified the sentences in a source document \(D_S\) that are most closely related to the sentences in a (potential) plagiarized document \(D_P\), we determine the overall percentage of plagiarism of \(D_P\) with respect to \(D_S\) as

\[
\text{Resem}(D_P, D_S) = \frac{\sum_{i=1}^{|D_P|} \text{SenSim}(P_i, D_S)}{|D_P|}
\]

(4)

where \(|D_P|\) is the number of sentences in \(D_P\), and \(\text{Resem}(D_P, D_S) \neq \text{Resem}(D_S, D_P)\), if \(D_P \neq D_S\).

By averaging the computed \(\text{SenSim}\) values of sentences in \(D_P\), SimPaD determines the ratio of the (segments of)

**4 Experimental Results**

In this section, we introduce the datasets used for conducting an empirical study on SimPaD and present several evaluation measures for analyzing the performance of SimPaD in detecting (non-)plagiarized documents.

### 4.1 Datasets

In assessing the performance of SimPaD, we used two plagiarism corpora. The first one, denoted Webis-PC, is the Bauhaus University Plagiarism Corpus Webis-PC-08 [18], which consists of 101 original English documents downloaded from the ACM digital library (http://portal.acm.org/dl.cfm). There is a plagiarized version for each original document \(D\), which was generated by (i) including exact paragraphs in \(D\), (ii) excluding some sentences from \(D\), and/or (iii) adding sentences with words similar to the ones in \(D\). The second corpus, the Meter Corpus [2], was constructed as part of the Measuring Text Reuse Project at the University of Sheffield in U.K. (The Meter corpus) consists of 265 unique stories provided by the British Press Association (PA) that were clustered into two different subject areas: entertainment and law/court reporting, which were collected from July 1999 to June 2000. For each of the 265 stories, Meter provides one or more (non-)derived newspaper articles, which translates into 944 pairs of news articles published in a variety of newspapers such as The Sun, Daily Mirror, Daily Mail, etc. Each news article pair in Meter is classified as wholly-derived (i.e., when the PA stories are copied/paraphrased entirely), partially-derived (i.e., when PA is the major source used for writing a news article), and non-derived (i.e., when PA is not the original source). In rewriting news articles based on PA stories, the authors of [2] observe that common rewriting strategies include (i) using the exact content from a source sentence, (ii) paraphrasing text from the source story to report the same information, and (iii) including new text, i.e., reporting PA stories using a different context. In evaluating the performance of SimPaD using Meter, wholly- and partially-derived articles are treated as plagiarized [14].

To the best of our knowledge, besides Webis-PC and Meter, no other benchmark datasets are available for evaluating the performance of a plagiarism-detection approach.

---

\(^5\) According to [2], PA is the most prestigious press agency in the U.K., which provides news to 86 different national newspapers, as well as 470 radio and television broadcasts.

\(^6\) Non-derived news articles refer to publications that report (but do not plagiarize) the 265 stories provided by PA.
4.2 Resemblance Values

As discussed in Section 3, in comparing any two documents SimPaD computes their Resem values. Figure 2(a) (2(b), respectively) shows the Resem value of each of the 101 plagiarized documents (944 pairs of articles, respectively) and its corresponding source in Webis-PC (Meter, respectively). As shown in Figure 2(b), partially-derived news article pairs have a lower degree of resemblance than the wholly-derived news article pairs, but a higher degree of resemblance than non-derived pairs in Meter. Based on the Resem values shown in Figure 2, we observe that SimPaD adequately detects the proportion of content shared by documents, i.e., the percentage of plagiarism found in the (potentially) plagiarized documents.

4.3 A Threshold for Plagiarism Detection

Prior to determining the accuracy of SimPaD in detecting (non-)plagiarized documents, we set an appropriate threshold value, ResemTH, for automatically labeling a (non-)plagiarized document DP of a source document DS using the Resem(DP, DS) value.

In defining ResemTH, we used the ID3 implementation of the decision tree, since ID3 is commonly used for inductive inference based on a given training set of instances and is an effective method for classification [11]. We randomly selected 40 (60, respectively) documents from Webis-PC (Meter, respectively) and their corresponding plagiarized (non-)plagiarized, respectively) version, which yield 100 training instances (i.e., document pairs) for constructing the decision tree. Each training instance includes an attribute, which contains a Resem value, and the class value of the instance previously set (i.e., (non-)plagiarized as defined in Meter and plagiarized for Webis-PC). Using the constructed decision tree, a document DP is classified as a plagiarized version of DS, if Resem(DP, DS) ≥ 0.27, i.e., the ResemTH value.

4.4 Performance Evaluation

Using the established ResemTH value and the computed Resem values of the document pairs in Webis-PC and Meter, we evaluated the Accuracy (= #of Correctly Classified Documents / |corpus|) of SimPaD in correctly detecting (non-)plagiarized documents and the ErrorRate (= 1 - Accuracy) for misclassification, where |corpus| is the total number of document pairs in a corpus. As shown in Figure 3, in detecting the plagiarized documents in Webis-PC, SimPaD yields 100% accuracy and classifies the (non-)plagiarized news article pairs in Meter with a 96.2% accuracy rate. Note that none of the wholly-derived and non-derived news article pairs in Meter were misclassified, and of the thirty-six misclassified news article pairs (3.8% of the total number of 944 classified pairs) in the partially-derived category (with a total of 438 news article pairs), each of its plagiarized copy yields a Resem value lower than ResemTH due to the small size of its corresponding news article, which includes only 2 to 4 sentences and only half of these sentences are (partially) derived from the corresponding PA source article. Even though SimPaD misclassified 3.8% of the articles in Meter as non-plagiarized, which are false negatives, SimPaD did not generate any false positives, i.e., all of the non-plagiarized articles were correctly identified.

4.5 Comparing SimPaD’s Performance

In order to further assess the effectiveness of SimPaD in detecting (non-)plagiarized documents, we compare its performance, in terms of accuracy, with other existing plagiarism-detection approaches, whose performance evaluations are based on Meter. (None of the performance evaluations of existing plagiarism-detection methods are based on Webis-PC, which is relatively new).

The plagiarized-detection method proposed by [14] uses a binary (i.e., similar and non-similar) classifier based on style features, such as frequent words in a document, and vocabulary features, i.e., tf * idf-weighted vectors of unigrams, in a given document to identify copyright infringement. The combined approach yields an accuracy of 70.5% in detecting (non-)plagiarized news articles out of the 88 selected pairs in Meter. In addition, [1] propose using the overlapping between n-grams in any two documents to determine the proportion of shared content. Experimental results conducted on wholly-derived and non-derived, law/court news article pairs in Meter report an overall 74%
accuracy rate [1]. As shown in Figure 3, SimPaD outperforms the two approaches, which are the only ones that we could find based on Meter for performance evaluation.

5 Conclusions

We have proposed a plagiarism-detection method, SimPaD, which relies on pre-computed word-correlation factors for determining the sentence-to-sentence similarity and eventually the degree of resemblance of any two documents to detect the plagiarized copy. SimPaD, which can handle various plagiarism techniques based on substitution, addition, and deletion of words in sentences, as well as sentence splitting and merging, provides the users a visual representation of sentences in a given source document that are paraphrased in its plagiarized version. Experimental results show that SimPaD (i) achieves an average of 98% accuracy in detecting (non-)plagiarized documents using two different benchmark datasets, and (ii) outperforms existing plagiarism-detection approaches in terms of accuracy by a huge margin, which verify the effectiveness of SimPaD in identifying (non-)plagiarized documents.

References


Figure 2. Resem values of documents and their (non-)plagiarized versions computed by SimPaD