Countering Plagiarism by Exposing Irregularities in Authors Grammars

Michael Tschuggnall and Günther Specht
Databases and Information Systems
Institute of Computer Science, University of Innsbruck, Austria
{michael.tschuggnall, guenther.specht}@uibk.ac.at

Abstract—Unauthorized copying or stealing of intellectual properties of others is a serious problem in modern society. In case of textual plagiarism, it becomes more and more easier to find appropriate sources using the huge amount of data available through online databases. To counter this problem, the two main approaches are categorized as external and intrinsic plagiarism detection, respectively. While external algorithms have the possibility to compare a suspicious document with numerous sources, intrinsic algorithms are allowed to solely inspect the suspicious document in order to predict plagiarism, which is important especially if no sources are available. In this paper we present a novel approach in the field of intrinsic plagiarism detection by analyzing syntactic information of authors and finding irregularities in sentence constructions. The main idea follows the assumption that authors have their mostly unconsciously used set of how to build sentences, which can be utilized to distinguish authors. Therefore the algorithm splits a suspicious document into single sentences, tags each word with part-of-speech (POS) classifiers and creates POS-sequences representing each sentence. Subsequently, the distance between every distinct pair of sentences is calculated by applying modified sequence alignment algorithms and stored into a distance matrix. After utilizing a Gaussian normal distribution function over the mean distances for each sentence, suspicious sentences are selected, grouped and predicted to be plagiarized. Finally, thresholds and parameters the algorithm is optimized by applying genetic algorithms. The approach has been evaluated against a large test corpus of English documents, showing promising results.

I. INTRODUCTION

Today more and more text documents are made publicly available through large text collections or literary databases. As recent events show, the detection of plagiarism in such systems becomes considerably more important as it is very easy for a plagiarist to find an appropriate text fragment that can be copied, where on the other side it becomes increasingly harder to correctly identify plagiarized sections due to the huge amount of possible sources.

The two main approaches for detecting plagiarism in text documents are external and intrinsic methods, respectively. Given a suspicious document, external algorithms are allowed to compare text fragments with any available sources (e.g. collections from the world wide web), whereas intrinsic algorithms try to detect plagiarism by inspecting the suspicious document only. Frequently applied techniques in both areas as well as in the related topics like authorship identification or text categorization include n-gram comparisons [11], [2], standard IR techniques like common subsequences [3], or complexity analysis [15], [4] combined with machine learning techniques. Especially in the fields of intrinsic plagiarism detection and authorship attribution, document-internal information that comprises the writing style of authors like the structure of words and/or sentences [22] or the occurrence of spelling errors [7] has to be taken into account additionally as such information is substantial when there are no comparable sources available.

In this paper we present a novel technique in the field of intrinsic plagiarism detection which analyzes the grammar of authors and utilizes the extracted features with common genetic sequence alignment algorithms. The approach is based on the assumption that authors differ in their writing style, i.e. in the way they construct their sentences. For example, the sentence

1

(1) The strongest rain ever recorded in India shut down the financial hub in Mumbai, officials said today.

could also be formulated as

(2) Today, officials said that the strongest Indian rain which was ever recorded forced Mumbai’s financial hub to shut down.

which is semantically equivalent but differs significantly according to its syntax. According to this assumption (which has been shown to be useful in plagiarism detection systems [22], [24]) we use ordered part-of-speech (POS) tags to represent the construction of sentences, ignoring information of concrete word-usage. Then, POS tag sequences are created for each sentence, whereby each sequence is compared with each other by sequence alignment and the results are stored into a distance matrix. Finally, using a Gaussian normal distribution sentences that differ significantly from each other according to their building structure are marked as suspicious.

The primary intent of this work is to investigate the applicability of POS tags and the corresponding algorithms in the field of intrinsic plagiarism detection systems. Therefore it does not consider the combination with other existing approaches but shows the potential of grammar analysis to counter frauds in text documents instead.

The rest of this paper is organized as follows: In Section II we propose an algorithm to expose plagiarism in text documents by using POS tags and sequence alignment. Genetic optimization algorithms are applied in Section III to optimize

---

1example taken and modified from the Stanford Parser website [19]
the parameters and thresholds used. After showing an extensive evaluation of the approach at the end of the section, related work and a conclusion is presented in Section IV and Section V, respectively.

II. THE POS-PLAGINn ALGORITHM

The aim of the POS-PlagInn\(^2\) algorithm is to intrinsically identify plagiarism in text documents. Thereby the main idea is to analyze the writing style of authors with respect to the grammar used, and to distinguish normal sentences from suspicious ones according to their syntactic building structure. This is done by storing distances between sentences in a distance matrix and then inspecting their average distances to other sentences. To calculate the distance of two sentences, modified dynamic programming algorithms which are used in Genetics, have been adapted to align sequences of POS tags. Finally, the sentence selection algorithm proposed in [23] is used to identify suspicious sections.

Concretely, the algorithm consists of the following steps:

A) Cleaning the document (e.g. remove unnecessary whitespace or non-alphanumeric characters) and splitting it into single sentences.
B) Calculating POS tags for each extracted sentence and computing POS tag sequences reflecting the building structure.
C) Comparing each POS tag sequence with each other sequence and storing the difference into a distance matrix.
D) Calculating the average distance for each sentence and utilizing a Gaussian normal distribution function.
E) Using the sentence selection algorithm and mark sentences/text sections as suspicious that differ significantly.

In the following the different parts are described in detail.

A. Document cleaning

At first the document is preprocessed by eliminating unnecessary whitespaces or non-parsable characters. For example, in global authorship attribution or plagiarism detection training and test sets often used document types are based on novels and articles of various authors, whereby sometimes even OCR text recognition is used due to the lack of digital data. Additionally, such documents contain problem sources like erroneous whitespaces or non-parsable characters. For example, in the sentence (1) of Section I is tagged with the following tags:

\[
\text{The/DT strongest/JJS rain/NN ever/RB recorded/VBN in/IN India/NNP shut/VBD down/RP the/DT financial/JJS hub/DT in/IN Mumbai/NNP, officials/NNS said/VBD today/NN.}
\]

Thereby the tags correspond to grammatical building constructs, e.g. DT indicates a determiner, JJS a superlative adjective or NN a noun. Although the information extracted by performing POS tagging only is more shallow than constructing the whole syntax grammar tree, results show that it is sufficient to find inconsistencies in text building structures (we proposed a plagiarism detection algorithm that incorporates syntax grammar trees in [23]). Moreover it is computationally significantly easier, especially with longer and more complex sentences.

In order to reflect the structure information only, the concrete word information is neglected. Thus, for each sentence a POS tag sequence like the following is created and stored:

\[
\text{DT-JJS-NN-RB-VBN-IN-NNP-VBD-RP-DT-JJ-DT-IN-NNP-NNS-VBD-NN}
\]

C. Distance calculation

Now, having all POS tag sequences for all sentences of the suspicious document, each sequence is compared with each other. The idea is to quantify the difference of every distinct POS sequence pair and to store its difference in a distance matrix \(D_n\):

\[
D_n = \begin{pmatrix}
    d_{1,1} & d_{1,2} & \cdots & d_{1,n} \\
    d_{2,1} & d_{2,2} & \cdots & d_{2,n} \\
    \vdots & \vdots & \ddots & \vdots \\
    d_{1,n} & d_{2,n} & \cdots & d_{n,n}
\end{pmatrix} = \begin{pmatrix}
    0 & d_{1,2} & \cdots & d_{1,n} \\
    * & 0 & \cdots & d_{2,n} \\
    \vdots & \vdots & \ddots & \vdots \\
    * & \cdots & * & 0
\end{pmatrix}
\]

The individual values of the matrix correspond to the difference between two sentences, i.e. their POS tag sequences. E.g. \(d_{1,3}\) corresponds to the distance between sentence 1 and 3 of the same document. As the distance between sentence \(i\) and \(j\) is the same as the distance between sentence \(j\) and \(i\), the resulting distance matrix is triangular. Hence, filling \(D_n\) requires only \(\binom{n}{2} = \frac{n(n-1)}{2}\) distance calculations, where \(n\) corresponds to the total number of sentences of a document.

The distances themselves are calculated using modified sequence alignment algorithms that are frequently applied in the fields of Genetics. More concretely, the well-known algorithms Global Alignment (Needleman-Wunsch) [10] and Local Alignment (Smith-Waterman) [16] have been adapted to align POS sequences. The original idea of the algorithms is to ideally align two DNA sequences using cost models that incorporate gene (mis)matches, insertions and deletions, resulting in an overall score of how similar the sequences are. Global alignments solutions align the whole sequences, whereas the local alignment algorithm searches for a local maximum of similarity. For example, using 2 for a match score and -1 for insertion, deletion or mismatch costs, respectively.

\[^2\text{POS-PlagInn is a variation of the intrinsic plagiarism detection algorithm PlagIn [23] (standing for Plagiarism and Innsbruck)}\]

\[^3\text{Apache OpenNLP. http://incubator.apache.org/opennlp, visited January 2013}\]
the similarity value of the DNA-strings CTCTAGCATT\textsuperscript{4} and GTGCAC could result in a score of 2 for the local alignment
\[
\begin{align*}
\text{C} & \quad \text{T} & \quad \text{T} & \quad \text{A} & \quad \text{G} & \quad \text{C} & \quad \text{A} & \quad \text{T} & \quad \text{T} \\
| & & & & & & & & \\
& & & & \text{G} & \quad \text{T} & \quad \text{G} & \quad \text{C} & \quad \text{A} & \quad \text{C} \\
\end{align*}
\]
and a score of 7 for the local alignment
\[
\begin{align*}
\text{T} & \quad \text{A} & \quad \text{G} & \quad \text{C} & \quad \text{A} \\
| & & & & & & & & \\
& & & & \text{T} & \quad \text{G} & \quad \text{C} & \quad \text{A} \\
\end{align*}
\]
In case of the plagiarism detection algorithm the alignment methods have been modified to be able to align POS tags. Thereby each tag is treated as a single instance: for example, aligning the two sentences
(1) \textit{This/DT is/VBZ a/DT simple/JJ sentence/NN}
(2) \textit{This/DT is/VBZ not/RB the/DT most/RBS complex/JJ construction/NN}

using the same scoring scheme would result in the following global alignment of score 8:
\[
\begin{align*}
\text{DT} & \quad \text{VBZ} & \quad \text{-} & \quad \text{DT} & \quad \text{-} & \quad \text{JJ} & \quad \text{NN} \\
| & & & & & & \\
& & & & \text{DT} & \quad \text{VBZ} & \quad \text{RB} & \quad \text{DT} & \quad \text{RBS} & \quad \text{JJ} & \quad \text{NN} \\
\end{align*}
\]
In order to not only incorporate the maximum local similarity calculated by the Local Alignment algorithm but to reflect the difference of whole sentences, we chose to use the global alignment algorithm as a first attempt. Additionally global alignment scores implicitly also incorporate differences in sentence lengths, a feature which has also been used in previous approaches. Nevertheless, manual analysis of the evaluation of some random documents indicates that there is no significant difference in overall results when using Global or Local Alignment.

As the POS-PlagInn algorithm tries to find inconsistencies rather than similarities between sentence structures, the scoring scheme is inverted, i.e. using a negative value for matches and positive values for mismatches, insertions or deletions, respectively. The weights for all further optimizations and evaluations have preliminary been defined to be -1 for matches, 1 for insertions/deletions and 1 for mismatches. With the inverted scheme, the distance matrix \(D_n\) as stated above is calculated by aligning each pair of sentences \(i\) and \(j\) (i.e. their POS tag sequences) and storing their distance \(d_{i,j}\).

An example of a visualized distance matrix of a document containing about 2000 sentences is illustrated in Figure 1. To provide better visibility, the triangular character of the distance matrix has been ignored in this case. The \(z\)-axis represents the POS tag sequence distance between the sentences on the \(x\)- and \(y\)-axis, and it can be seen that there are significant differences in the style of the text around sentence number 900. Additionally it is important to note that these distances are significantly different with respect to all other sentences, i.e. they are not just local peaks.

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{distance_matrix.png}
\caption{Distance Matrix of a Sample Document Consisting of about 2000 Sentences with Global Peaks.}
\end{figure}

D. Finding significant differences

According to what can already be estimated by inspecting the visualized distance matrix with the human eye, the task of the following step is to computationally find the sentences which differ significantly. To find significantly outstanding sentences, i.e. sentences that might have been plagiarized, the median distance for each row in \(D\) is calculated. The resulting vector \(\mathbf{d} = (d_1, d_2, ..., d_n)\) is then fitted to a Gaussian normal distribution which estimates the mean value \(\mu\) and the standard deviation \(\sigma\). The two Gaussian values can thereby be interpreted as a common variation of how the author of the document builds his sentences grammatically.

Finally, all sentences that have a higher difference than a predefined threshold \(\delta_{\text{susp}}\) are preliminary marked as suspicious. The definition and optimization of \(\delta_{\text{susp}}\) (where \(\delta_{\text{susp}} \gg \mu + \sigma\)) is shown in Section III-B. Figure 2 depicts the mean distances resulting from averaging the distances for each sentence in the distance matrix \(D\). After fitting the data to a Gaussian normal distribution, the resulting mean \(\mu\) and standard deviation \(\sigma\) are marked in the plot. The threshold \(\delta_{\text{susp}}\) that splits ordinary from suspicious sentences can also be seen, and all sentences exceeding this threshold are marked.

E. Sentence Selection

In the last step, sentences preliminary marked as suspicious are grouped into sections in order to identify whole text paragraphs rather than single sentences. By using the algorithm proposed in [22] sentences are thereby either grouped into text sections or unmarked, i.e. marked as non-suspicious. The main idea of the technique used is to traverse all (preliminary marked and not marked) sentences of the document and to perform the following tasks:

- Group adjacent suspicious sentences into suspicious text sections.

\textsuperscript{4}The characters represent the nucleotides contained in a DNA string: Adenine (A), Guanine (G), Cytosine (C), Thymine (T).
ARAMETER  

\[ \delta_{\text{true}} = \delta = 0 \]

filterSingles  

\[ \text{maxLookahead} \]

is used to determine and different

Inspect suspicious sections that contain only one sen-

“At what time?”

Use a lookahead variable to also take non-suspicious

sections into account that reside between suspicious sections. If such sentences are found before the al-

algorithm reaches the predefined maximum maxLooka-

head, they are also added to the suspicious section.

As the presented algorithm is based on the gram-

matical structure of sentences, short instances like

“I like tennis.” or “At what time?” carry too less

information and are often not marked as suspicious as their building structure is too simple. Nevertheless,
such sentences may be part of a plagiarized section and should therefore be detected. For example, if eight sentences in a row have found to be suspicious except one in the middle, it is intuitively very likely that it should be marked as suspicious as well. Previous evaluations showed that this step leads to better results and therefore proved the intuitive assumption that when authors plagiarize, they do so for whole sections rather than for single sentences [22].

Inspect suspicious sections that contain only one sen-
tence and decide whether this section should be kept or discarded. The decision is thereby based on the additional threshold parameter \( \delta_{\text{single}} \), which defines the minimum amount of how different a sentence structure must be in order to not be discarded for the final result. In other words, a standalone suspicious sentence has to be very different in order to remain suspicious.

Figure 3 illustrates the selection of sentences for the final result, taken from a different sample file. The left side (a) shows the sentences and their marking as suspicious and non-
suspicious as the result from the previous step (Section II-D). Suspicious sentences are highlighted, which is the case for all sentences having a mean distance greater than \( \delta_{\text{susp}} \). On the right side (b) the final result of the algorithm is shown, indicating three plagiarized sections. It can be seen that some non-suspicious sentences have been included, where on the other hand one single-sentence suspicious section has been excluded.

III. PARAMETER OPTIMIZATION AND EVALUATION

A. Test corpus

To evaluate and optimize the algorithm, the PAN 2011 training and test corpora [14] have been used which contain over 4000 English documents, respectively. The documents consist of a various number of sentences, starting from short texts from e.g. 50 sentences up to novel-length documents of about 7000 sentences. About 50\% of the documents contain plagiarism, varying the amount of plagiarized sections per document, while the other 50\% are left originally and contain no plagiarized paragraphs.

Most of the plagiarism cases are built by copying text fragments from other documents and subsequently inserting them in the suspicious document, while manual obfuscation of the inserted text is done additionally in some cases. Also, some plagiarism cases have been built using copying and translating from other source-languages like Spanish or German. Finally, for every document there exists a corresponding annotation file which can be consulted for an extensive evaluation.

B. Parameter Optimization

As shortly described in the previous section, the sentence-

selection algorithm relies on various input variables:

- \( \delta_{\text{susp}}' \): suspicious sentence threshold. Every sentence that has a higher mean distance is marked as suspicious.

- \( \delta_{\text{single}} \): single suspicious sentence threshold. Every sentence in a single-sentence plagiarized section that is below this threshold is unmarked in the final step of the algorithm.

- maxLookahead: maximum lookahead. Defines the maximum value of checking if there is a suspicious sentence occurring after non-suspicious sentences that can be included into the current plagiarized section.

- filterSingles: boolean switch that indicates whether sections containing only one sentence should be filtered out. If filterSingles = true, the single suspicious sentence threshold \( \delta_{\text{single}} \) is used to determine whether a section should be dropped or not.

Thereby, the values for the thresholds \( \delta_{\text{susp}}' \) and \( \delta_{\text{single}} \) respectively, represent the inverse probability range of the Gaussian curve that include sentences with a mean distance that is not marked as suspicious. For example, \( \delta_{\text{susp}}' = 0.9973 \) would imply that \( \delta_{\text{susp}} = \mu + 3\sigma \), meaning that all sentences
having a higher average distance than $\mu + 3\sigma$ are marked as suspicious. In other words, one would find 99.73% of the values of the Gaussian normal distribution within a range of $\mu \pm 3\sigma$. In Figure 2, $\sigma_{\text{susp}}$ resides between $\mu + 2\sigma$ and $\mu + 3\sigma$.

In the following the optimization techniques are described which should help finding the parameter configuration that produces the best result. To achieve this, genetic algorithms have been used on the training set. Additionally, they have been applied to find optimal configurations on two distinct training document subsets that have been split by the number of sentences (see Section III-B2).

All configurations have been evaluated using the common IR-measures recall, precision and the resulting harmonic mean $F$-measure. In this case recall represents the percentage of plagiarized sections found, and precision the percentage of correct matches, respectively. In order to compare this approach to others, the algorithm defined by the PAN workshop [14] has been used to calculate the according values. Furthermore the PAN workshop metric granularity has been measured. It describes the "grouping" quality of correctly annotated plagiarized sections, meaning that a continuous plagiarized section should also be annotated as one section rather than more sections. For example, a granularity value of 2 would denote that the algorithm in average predicted 2 sections instead of one. Hence, a granularity value of 1 is optimal.

1) Genetic Algorithms: An intuitive attempt to find an optimal parameter configuration would be to define value ranges for each parameter and test every configuration for a maximum outcome. Since the document test set is rather large, especially the parsing, POS tagging and alignment parts are computationally intense. Moreover, by just using predefined parameter configurations, the optimal solution may not be found as it may consist of unexpected parameter combinations. Therefore to find optimal configurations, genetic algorithms have been used which are designed to find at least individual local maxima without restricting the value ranges of parameters.

Genetic algorithms emulate biological evolutions and implement the principle of the "Survival of the fittest". In this sense genetic algorithms are based on genes (the parameters) and chromosomes (the parameter configurations). By evolving successful populations of chromosomes iteratively, genetic algorithms are likely to find (at least local) optima. Using the JGAP-library which implements genetic programming algorithms, the parameters of the sentence-selection algorithm have been optimized. As the algorithm still needs a high amount of computational effort and to avoid overfitting, random subsets of 1000 to 2000 documents have been used to evaluate each chromosome, whereby these subsets have been randomized and renewed for each evolution.

The best configuration reaching an $F$-measure of about 30% can be seen in Table I where $p$ and $l_{\text{max}}$ correspond to population sizes and the maximum lookahead used, respectively. Interestingly, what can be seen as well is that the best configuration produced by a population size of 400 recommends to filter single-sentence plagiarism sections with a lower threshold than multiple-sentence sections. Although a better configuration could be found, this result still represents a local maximum which indicates that authors might as well plagiarize single sentences rather than longer text fragments only.

2) Genetic Algorithms On Document Subsets: By a manual inspection of the individual results for each document of the test corpus it could be seen that in some configurations the algorithm produced very good results on short documents, while on the other hand it produced poor results on longer, novel-length documents. Controversially, other configurations produced the vice versa result.

Therefore, to make use of the assumption that different length documents should be treated differently, the test corpus has been split by the number of sentences in a document. For example, when using 150 as splitting number, the subsets $S_{<150}$ and $S_{\geq 150}$ have been created, containing all documents that have less than 150 sentences and containing all documents that have more than or exactly 150 sentences, respectively (150 sentences is approximately then size of a full paper). Then, for each of the two subsets, the optimal parameter configuration has been evaluated using genetic algorithms as it is described in Section III-B1. Like before, a random number of documents from 1000 to 2000 has been used to evaluate a chromosome.

Table II shows the best configurations produced by genetic algorithms using the sentence-split document subsets. For the dividing numbers 100, 150 and 200 sentences per document have been choosen.

With an $F$-measure of about 52% for documents having less than 100 sentences, and about 22% for all longer documents, respectively, a total $F$-value of 37% could be achieved with the best parameter configuration. Thereby the algorithm performed significantly better on documents having less sentences on all three evaluation parts, which indicates that it works very good on about paper-length sizes and has drawbacks on novel-length sizes. A first intuitive conclusion may be that the longer documents get, the more variety on sentence constructions it contains, leading to blurring syntax.

The granularity for both small and large documents is near to or equal to optimum, hence the document size seems to be irrelevant in case of granularity measures. While the thresholds and lookaheads vary for the different subsets, all configurations except one recommend to filter out single-sentence plagiarized sections with the respective threshold.

As a general remark, the dividing numbers of 100, 150 and 200, respectively, have been chosen manually and are not guaranteed to be optimal. In this sense future work should
also incorporate genetic algorithms to find the optimal dividing threshold.

C. Evaluation

Using the document sets of the PAN 2011 workshop as described in Section III-A, the algorithm has been trained and optimized with the training set and subsequently evaluated against the test corpus consisting of about 4000 English documents.

Table III summarizes the evaluation results gained from using the optimization techniques described earlier. With an F-score of 52% the best result could be achieved by applying the parameter setting optimized for documents having less than 100 sentences. While this score has been found by evaluating the corresponding document subset only (<100 sentences), an F-score of 30% could be reached by using a single parameter configuration over the whole test set.

Regardless of the optimization splitting number of sentences to be contained in a document, the algorithm works best for the document subsets having fewer sentences. While the subsets with documents having more sentences achieve only an F-score of about 10-20%, their counterpart reside with F-scores of about 50%. By combining the individual results for splitted subsets, F-scores of about 35% could be reached. Concretely, the variant using the splitting number of 100 sentences per document worked best with an F-score of 37%. This is also the global optimum in comparison to the evaluation using just a single parameter setting, which achieves an F-score of about 30%. Generally, recall and precision values are balanced in all settings, tending to have a higher precision in most cases.

Overall, the results are very promising as other current intrinsic plagiarism detection approaches reside with an F-score of about 33% [12], [13]. Compared to previous experiments using whole syntax grammar trees achieving about 35% in the best setting this approach performs even better.

In order to avoid presenting overfitted results adjusted to the training set, the variant using just a single parameter configuration over the whole test subset has been used for all further analyses. Therefore, because previous optimizations already showed that the algorithm is sensitive on the number of sentences per document, the single parameter configuration has been evaluated according to number of sentences. The results illustrated in Figure 4 indicate that the less sentences a document has, the better the algorithm operates, i.e. predicts plagiarism correctly. For documents consisting of less than 50 sentences, an F-score of over 65% could be reached. On the other side, a possible explanation for the rather poor results concerning longer, novel-length documents might be that structural syntactic differences are blurred on longer documents. In other words, if an author writes long documents, s/he may also use different sentence construction sets throughout the document in an unconscious or even conscious manner that compensate differences in style.

To investigate the results for documents containing plagiarism, the number of false-positives and false-negatives, respectively, have been evaluated. The results depicted in Figure 5 show that the number of false-positives is rather high, i.e. the algorithm often predicts plagiarism where there actually is none. Hence, the algorithm should be improved in the future by eliminating false-negatives.

Diagram (a) in Figure 5 illustrates the number of falsely detected plagiarized sections on plagiarism-free documents, and the number of not detected plagiarized sections on documents that contain plagiarism, where the number of false-positives is significantly higher. According to previous observations the rate of false-positives has been compared with the number of sentences contained in a document. The result shown in diagram (b) make clear that the main set of false-positives result from long documents, i.e. the shorter the document is, the less the probability of falsely predicted plagiarism is. Finally diagram (c) states that the algorithm predicts too much plagiarism cases in general. For example, if a document contains three plagiarized sections, it is often the case that significantly more than three sections are predicted, which results in a decrease of the F-score for documents containing plagiarism.

The number of documents where the algorithm predicted too little and the number of documents where it predicted the exact amount of plagiarism cases are balanced equally. Exact predictions in this case do not necessarily correspond to correct predictions, e.g. it might be the case that a document contains one plagiarized section and that also one section is predicted by the algorithm, but the two sections are disjunct. Nevertheless, manual inspections of individual results reveal that this scenario is very rare.

IV. RELATED WORK

An often used technique in intrinsic plagiarism detection algorithms are n-grams [17], [6], where a text document is broken into sets of two-, three- or four-letter chunks and

<table>
<thead>
<tr>
<th>document subset</th>
<th>Granularity</th>
<th>Recall</th>
<th>Precision</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>less than 100 sentences</td>
<td>1.0</td>
<td>0.520</td>
<td>0.524</td>
<td>0.522</td>
</tr>
<tr>
<td>less than 150 sentences</td>
<td>1.0</td>
<td>0.504</td>
<td>0.527</td>
<td>0.515</td>
</tr>
<tr>
<td>less than 200 sentences</td>
<td>1.0</td>
<td>0.475</td>
<td>0.495</td>
<td>0.485</td>
</tr>
<tr>
<td>all documents</td>
<td>1.09</td>
<td>0.282</td>
<td>0.329</td>
<td>0.304</td>
</tr>
</tbody>
</table>

TABLE III. BEST EVALUATION RESULTS USING THE PAN IT DOCUMENT TEST SET.
subsequently analyzed by their number of occurrences within sliding windows. With the additional use of a style change function [17] could reach a recall and precision value of about 33% over the PAN11 test corpus. Another approach also uses the sliding window technique but is based on word frequencies and the assumption that authors use a significant set of words [12].

A similar approach like the one proposed in this paper also analyzes the syntax of sentences, but computes whole grammar trees instead of POS tags only [23]. It calculates pq-grams [1] from grammar trees for each sentence and uses the respective pq-gram distance to estimate the tree edit distance between sentence grammar trees.

An approach that tries to recognize paraphrased sections based on the phrase structure of sentences and the structure of verb phrases is described in [24]. In this work, sentence-initial and –final phrases are inspected together with predefined semantic classes of verbs [8] by part-of-speech tagging. It uses POS-tags only, i.e. without referring to computationally intense full grammar tree parses.

[15] discusses the comparison of binary strings calculated from word groups like nouns or verbs using complexity analysis. Word-n-grams are also utilized in [2] to expose plagiarism in text documents. Approaches in the field of author detection and genre categorization also use NLP tools to analyze documents based on syntactic annotations [18]. Word- and text-based statistics like the average sentence length or the average parse tree depth are used in [5].

Another interesting approach used in authorship attribution that tries to detect the writing style of authors by analyzing the occurrences and variations of spelling errors is proposed in [7]. It is based on the assumption that authors tend to make similar spelling and/or grammar errors and therefore uses this information to attribute authors to unseen text documents.

V. Conclusion and Future Work

In this paper we presented a novel approach in the field of intrinsic plagiarism detection that uses grammar analysis of authors in order to predict plagiarism in text documents by inspecting the suspicious documents only. The main idea follows the assumption that different authors use different syntaxes to build their sentences, and that this information is significant enough to find plagiarized text fragments. To find irregularities in sentence construction syntaxes a suspicious document is split into single sentences, where subsequently all sentences are POS-tagged and compared against each other by applying modified sequence alignment algorithms. Finally a sentence selection algorithm using a Gaussian normal distribution function is used to find sentences and text sections that differ significantly compared to the rest of the text.

The approach has been evaluated against a large test collection of documents containing plagiarism-free and plagiarized English documents, showing that syntactic information can help to expose plagiarism. By applying several optimization techniques using genetic algorithms, the parameters and thresholds needed for the algorithm have been optimized resulting in an F-score of about 30%. By additionally optimizing the parameters for different document lengths, e.g. by using different parameter configurations for full-paper-length documents and novel-length documents, respectively, an F-score of over 52% could be reached. Compared to other approaches in the field of intrinsic plagiarism detection this result is very promising.
Many analyses of evaluation results show that the algorithm works well on short documents in general, but lacks of predicting the correct text sections that have been plagiarized especially on longer documents. Concretely, the algorithm predicts too many sentences to be plagiarized which results in a relatively high number of false-positives. Future work should further investigate in means to decrease this number in order to achieve even higher F-scores.

From a wider perspective it seems obvious that intrinsic plagiarism detection algorithms in general are a useful tool to (pre)check documents for plagiarism if there are no comparable sources available. Nevertheless, even with a rather high F-score of over 50% achieved with this approach for document subsets, external approaches reach significantly more than 90%, but can only be used if appropriate sources are available.

Furthermore, the approach presented in this paper is very suitable for the use in the field of author identification, where unseen text documents have to be attributed to authors. Future work should therefore try to adapt the algorithm to be usable for tasks like authorship attribution and verification, multi-author text clustering or even text genre categorization.

To improve the algorithm presented in this paper, future work should also incorporate and evaluate Local Alignment or even a combination of Global/Local Alignment for calculating distances between POS sequences. Additionally, the preliminary defined costs/benefits needed by the alignment algorithms could also be optimized, e.g. by applying genetic optimization algorithms presented in this paper.

Other possibilities to enhance the algorithm would be to combine it with other techniques that consider word-level information such as word categories, frequencies or character/word n-grams. Moreover, the algorithm might perform better on other languages which have a more complex grammar syntax such as German or French, as there are more possibilities to construct sentences and thus making the style of authors more unique.

REFERENCES


