On Divergence-based Author Obfuscation: An Attack on the State of the Art in Statistical Authorship Verification

Janek Bevendorff, Tobias Wenzel, Martin Potthast, Matthias Hagen, Benno Stein

Abstract: Authorship verification is the task of determining whether two texts were written by the same author based on a writing style analysis. Author obfuscation is the adversarial task of preventing a successful verification by altering a text's style so that it does not resemble that of its original author anymore. This paper introduces new algorithms for both tasks and reports on a comprehensive evaluation to ascertain the merits of the state of the art in authorship verification to withstand obfuscation. After introducing a new generalization of the well-known unmasking algorithm for short texts, thus completing our collection of state-of-the-art algorithms for verification, we introduce an approach that (1) models writing style difference as the Jensen-Shannon distance between the character n-gram distributions of texts, and (2) manipulates an author's writing style in a sophisticated manner using heuristic search. For obfuscation, we explore the huge space of textual variants in order to find a paraphrased version of the to-be-obfuscated text that has a sufficiently high Jensen-Shannon distance at minimal costs in terms of text quality loss. We analyze, quantify, and illustrate the rationale of this approach, define paraphrasing operators, derive text length-invariant thresholds for termination, and develop an effective obfuscation framework. Our authorship obfuscation approach defeats the presented state-of-the-art verification approaches, while keeping text changes at a minimum. As a final contribution, we discuss and experimentally evaluate a reverse obfuscation attack against our obfuscation approach as well as possible remedies.

ACM CCS: Applied computing → Document management and text processing; Applied computing → Computer forensic

Keywords: authorship verification, authorship obfuscation, privacy, computational ethics

1 Introduction

Can the authorial style of a text be consistently manipulated? More than a century worth of research on stylometry and authorship analysis could not produce a definitive answer to this question or a reliable approach to do so manually. In the context of computational authorship obfuscation, a handful of approaches have achieved some limited success but are still rather insufficient. Rule-based approaches are neither flexible, nor is stylometry understood well enough to compile rule sets that specifically target author style. For lack of a higher-level understanding and a generalizable framework for reliable deduction of an author's style from a given text, the state of the art in authorship verification has largely focused on the use of statistical methods for measuring the similarity of a text with other samples provided by the same or different authors. While simple in nature, the best approaches in the literature have turned out to be surprisingly robust and rather difficult to fool. The simplest text obfuscation attempts tend to result in sub-par text quality with often negligible effect, and even the more complex systems struggle with the same and other issues. For example, monolingual machine translation-based approaches suffer from a lack of training data, whereas applying multilingual translation in a cyclic manner as a workaround has proved to be ineffective. In addition, none of the existing approaches offers a means to control the result quality. Given recent advances in controlled text generation, it stands to reason that a lot more can be achieved.

In this paper, we introduce authorship verification by example of our own generalized adaptation of the *unmasking* algorithm developed by Koppel and Schler [33], and propose a novel obfuscation method that is able to successfully fool this and other state-of-the-art verification algorithms. In order to do so, we depart from the mentioned obfuscation paradigms and, for the first time, cast author obfuscation as a heuristic search problem. Given a to-be-obfuscated text, we search for a cost-minimum sequence of tailored paraphrasing operations that achieve a significant increase of the text's style distance to other texts from the same author under a generic statistical writing style representation; costs accrue through operations in terms of their estimated text quality reduction. By designing a hybrid search strategy, we obtain a significant reduction of the exponentially growing search space that is induced by the paraphrasing operators, enabling the use of informed search algorithms for authorship obfuscation.

Our key contributions are (1) a generalization of unmasking to short texts (Section 3), (2) a theory of style distance and length-dependent obfuscation, which is based on the Kullback-Leibler divergence (Section 4), and (3) an operationalization of this theory using heuristic search, which enables us to balance obfuscation gain and text quality loss (Section 5). In an extensive comparative evaluation, we benchmark the effectiveness of our obfuscation system against unmasking as well as a large number of other state-of-the-art authorship verification systems (Section 6). Finally, we investigate the vulnerability of our obfuscation approach to de-obfuscation attacks, where we attempt to undo an obfuscation by reversing the obfuscation procedure (Section 7). Parts of this work have been published before at other venues [3, 4] and are provided as an overview in the context of this special issue, whereas the inquiry into obfuscation safety or vulnerability is a novel contribution.

2 Related Work

Authorship analysis dates back over 120 years [9] and has mostly dealt with authorship attribution (given a text of unknown authorship and texts from known candidate authors, attribute the unknown text to its true author among the candidates). More recently, the task of authorship verification attracted a lot of interest (given a text of unknown authorship and a set of texts from one known author, verify whether the unknown text is written by that author) since it lies at the heart of many authorship-related problems.

Systematic reviews on authorship analysis have been contributed by Juola [23] and Stamatatos [46] and the effectiveness of character 3-grams today is "folklore knowledge," albeit not systematically proven. Still, a complete list of stylometric features has not been compiled to date. Abbasi and Chen [1] proposed *writeprints*, a set of over twenty lexical, syntactic, and structural text feature types, which has gained some notoriety within attribution, verification, but also for "anonymizing" texts [59, 37, 22, 35].

Instead of relying on a rich feature set, Zhao et al. [58], Zhao and Zobel [57] only extract POS tag and function word distributions and interpret style differences as measurable by the Kullback-Leibler divergence. The Kullback-Leibler divergence further appeared in an overview of style distance measures by Kocher and Savoy [32], yet in the context of authorship profiling, rather than verification. Teahan and Harper [54] and Khmelev and Teahan [28] use compression as an indirect means to measure stylistic difference; later adapted and improved by Halvani et al. [19]. Koppel and Schler [33] developed the *unmasking* approach. The basic idea behind this approach is that texts written by the same author only differ in few superficial features. By successively removing those superficial features, differentiability between texts by the same author is expected to degrade faster than for texts written by different authors. Unmasking has proved very successful particularly on longer texts [51, 27], although Sanderson and Guenter [44] demonstrated its particularly weak performance if the texts undercut a minimum length of about 5,000 words, which we will address as part of our core contributions. Authorship verification gained new traction from a series of dedicated shared tasks at PAN [24, 47, 49], which gave way to many new verification approaches—often specifically tailored towards short texts—establishing a stable baseline for the state of the art [15, 29, 2, 12]. Many new publications in the field have since appeared including both traditional statistical methods as well as deep transfer-learning approaches. However, none have managed to significantly push the boundaries of the state of the art on the task of short-text authorship verification, thus constituting an array of comparably competitive verification algorithms [32, 31, 7, 8, 20, 39]. The datasets used for the PAN shared tasks were later shown to incorporate a number of sampling-induced biases [5], warranting a closer re-examination of existing verification approaches.

Among the first to tackle authorship obfuscation were Rao and Rohatgi [42], who used cyclic machine translation where texts were automatically translated into various languages and then back to English. Following the publication of Koppel and Schler's unmasking algorithm, Kacmarcik and Gamon [26] developed a method that directly attacks unmasking. By iteratively removing the most discriminatory text features, the classification performance of an unmasking verifier could be degraded—at the cost of rather unreadable texts. Obfuscators targeting write-prints-based verifiers were later presented by Juola and Vescovi [25] and McDonald et al. [35]. Brennan et al. [10] found that machine translation for obfuscation is ineffective and due to its blackbox character also rather uncontrollable. Thus, instead of performing round-trip translation across multiple languages, Xu et al. [56] proposed within-language machine translation to translate directly between styles. The practicality of this approach, however, is diminished by the general lack of large-scale parallel training data, a limitation of neural approaches that still holds true today inhibiting the practicality of models like the ones proposed by Emmery et al. [13] or Bo et al. [6].

From 2016 to 2018, a shared task series on authorship obfuscation was organized at PAN [43, 18, 41]. Some of the seven participating teams suggested rather conservative rule-based approaches that did not change a text sufficiently to obfuscate authorship against most state-of-the-art verifiers. Other obfuscators "fooled" some of these verifiers, but yet again were generating rather unreadable texts. To score high in terms of text quality and obfuscation performance, the shared task organizers asked for approaches that more carefully *paraphrase* a text (i.e., the meaning should stay the same and the text should still be readable). Only recently, Mahmood et al. [34] presented Mutant-X, an obfuscator based on genetic algorithms that tries to tackle these problems while requiring less data than neural approaches. So far, this system has only been tested in a closed-set scenario with up to 10 authors and has not been compared against recent state-of-the-art authorship verifiers, but merits closer inspection in future work.

Our new obfuscation approach presented in this paper is inspired by Stein et al.'s [53] heuristic paraphrasing idea for "encoding" an acrostic in a text and by Kacmarcik and Gamon's observation that changing rather few text passages may successfully obfuscate authorship.

2.1 Unmasking for Authorship Verification

Unmasking as per Koppel and Schler is based on the idea that the style of texts from the same author differs only in a few superficial features. By iteratively removing these most discriminating style features, one can measure the "speed" at which cross-validation accuracy between sets of chunks of the two texts degrades. For texts written by the same author, the accuracy tends to decrease faster than otherwise. Combining the obtained accuracy values into curves for each pair, a meta classifier can be trained on the curves to determine the class of a pair (same / different author). Koppel and Schler evaluated their approach on a corpus of 21 books (each at least 500 kB) by 10 different authors. The task was to verify for each book A whether it has been written by a given author, using all the latter's books B for an author profile, except book A, in case it was the same author. As described in their paper, the unmasking algorithm works as follows (Figure 1):

1. From either text, create non-overlapping chunks of at least 500 words length without splitting paragraphs.
2. Use the 250 words with highest average frequency in A and B as features.
3. Obtain 10-fold cross-validation accuracy between A and B with a linear SVM kernel.
4. Eliminate the 3 highest positive and negative features for the model trained in each fold.
5. Go to Step 3 if there are features left.

The declining cross-validation accuracy values from curves on which a meta classifier is trained. Koppel and Schler used another SVM as the meta classifier, utilizing as features the curve points, the curves' point-wise first- and second-order derivatives, and the derivatives sorted by steepest point-wise drop. With this approach, they achieved a verification accuracy of over 95 %.

3 Unmasking Generalized to Short Texts

Unmasking as a strong state-of-the-art verification approach lends itself to serving as a good benchmark for our further inquiries into authorship obfuscation. Unfortunately, the performance of unmasking hinges on the availability of sufficiently many chunks per text, where each chunk has to be of at least the aforementioned 500 words length, or else the training data becomes too sparse and no descriptive curves can be generated. Short texts have the inherent problem that not many chunks can be extracted by cutting them into pieces. To generate more training samples from short texts, one method would be to generate overlapping chunks, but this only ends in many almost identical chunks and provides only a marginal performance boost. Instead, as a first contribution, we exploit the bag-of-words nature of the unmasking features and create the chunks by oversampling words in a bootstrap-aggregating manner.

We treat each text as a random pool of words from which we can draw without replacement to fill up a chunk. Once the pool is exhausted, we replenish it and draw again until we have generated a sufficient number of chunks. With this method, we essentially draw with replacement, but can guarantee that each word is drawn at least once.

Employing this bagging approach alone will not yield satisfying results, however. The curves will be quite random with high variance. To counteract this, we run unmasking on the generated chunks multiple times and average the curves to get smoother and more reproducible results. Our generalized unmasking algorithm works as follows (for brevity, we leave a hyperparameter discussion to Bevendorff et al. [4]):

1. From either text, create 30 chunks consisting of 700 words each by random chunk generation.
2. Use the 250 words with highest average frequency in A and B as features.
3. Obtain 10-fold cross-validation accuracy between A and B with a linear SVM kernel.
4. Eliminate the on average 5 most significant positive and negative features across folds (resulting in a total of 10 removals).
5. Go to Step 3 if there are still features left.

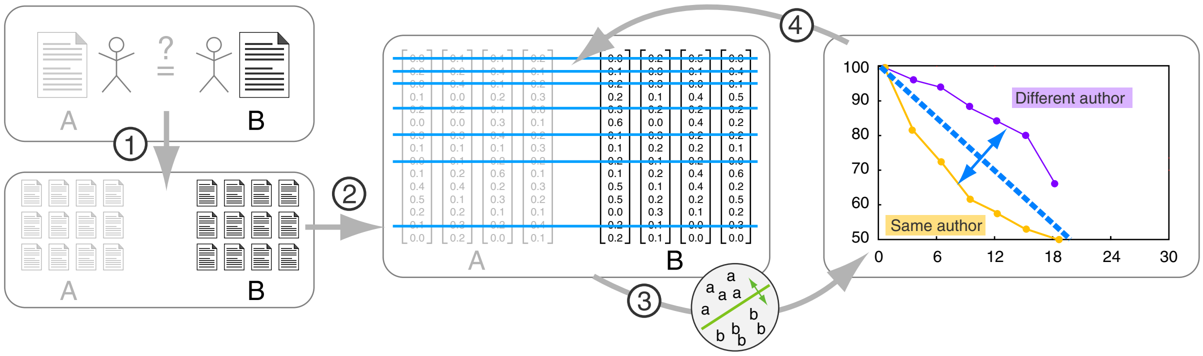


Figure 1: Schematic of the unmasking algorithm. Steps 1-4 are described in the text below. Dependent on whether the authors of texts A and B are the same or different, accuracy curves as exemplified can be expected.

Approach	Precision	Recall	$\mathbf{F_{0.5u}}$	c@1
Generalized Unmasking	0.82	0.54	0.74	0.76
Bagnall	0.81	0.71	0.77	0.79
Halvani et al. (CLM)	0.78	0.78	0.78	0.78
Halvani et al. (CBC)	0.71	0.71	0.71	0.70

Table 1: Comparison of generalized unmasking ($c = 0.1$) with the state of the art in short-text authorship verification. $F_{0.5u}$ is the $F_{0.5}$ measure with *unknown* as false negatives [4], c@1 is the non-answer-augmented accuracy as used for PAN [50]. Differences between the first three verifiers are non-significant.

Another linear SVM classifier is trained on these training curves, their central-difference gradients (first- and second-order), as well as their gradients sorted by magnitude. This meta classifier is then used to classify the curves that were generated in the same fashion from text pairs from the test set. As a means to add more control to the reliability of verification results obtained by unmasking, we further add a confidence hyperparameter c, which is the minimum distance to the meta classification hyperplane a generated unmasking curve must have in order to be classified. Any curve closer than this threshold is rejected as *unknown*. We verified the effectiveness of this new generalized unmasking approach by comparing its performance on the new Webis Authorship Verification Corpus 2019 [4, 5] of 262 authorship verification cases against the winning approach of the PAN 2015 Author Identification task by Bagnall [2] as well as a more recent compression-based approach by Halvani et al. [19]. The approach by Halvani et al. was tested with two of their proposed distance measures CLM and CBC. We found no significant differences between any of the three verification algorithms (with the exception of CBC performing worse overall), indicating a successful adaptation of Koppel and Schler's unmasking to short texts with state-of-the-art performance. Results are shown in Table 1.

4 Style Distance and Adaptive Obfuscation

Obfuscation and verification of authorship are two sides of the same coin. Consequently, effective obfuscation technology has to be developed from the verification perspective: Given texts from the same author, one of which is not publicly known to be written by that author, the goal is to paraphrase that text so that verification attempts against texts of known authorship will fail. The term "paraphrase" expresses that we aim at preserving the meaning of a text while maximizing its style distance regarding reference texts.

From theoretical and practical analyses it is known that current authorship verification algorithms analyze (be it implicitly or explicitly) the distributions of the character trigrams. We can state this without loss of generality, since character trigrams as the lowest-level feature that still captures word boundaries and morphology, directly influence higher-level features such as word n-grams or parts of speech. Although we only consider English in our research, this holds for at least most European languages. We exploit this fact by breaking the style comparison between two texts down to a single number, namely, the Kullback-Leibler divergence (KLD) of the respective character trigram frequency distributions. This effective character-based style distance measure has not been employed directly in this form for verification as of now although its mathematical properties are highly beneficial from a verification and obfuscation perspective: (1) The KLD acts as a feature- and task-agnostic information-theoretic divergence measure that directly or indirectly represents the core decision criterion not only for cross-entropy-based classifiers. (2) The KLD can be used as a simple and computationally feasible stopping criterion for an obfuscation process. (3) Based on the KLD, a normalization criterion for obfuscating texts of different lengths can be derived. (4) The KLD derivative can serve as a selection criterion for parts of the text that will yield the highest obfuscation gains if changed, immediately suggesting a greedy obfuscation algorithm.

4.1 Measuring Stylistic Distance

By utilizing character trigram frequencies to represent texts, one encodes various aspects of authorial style at the same time including vocabulary, morphology, and punctuation. Based on this representation, the

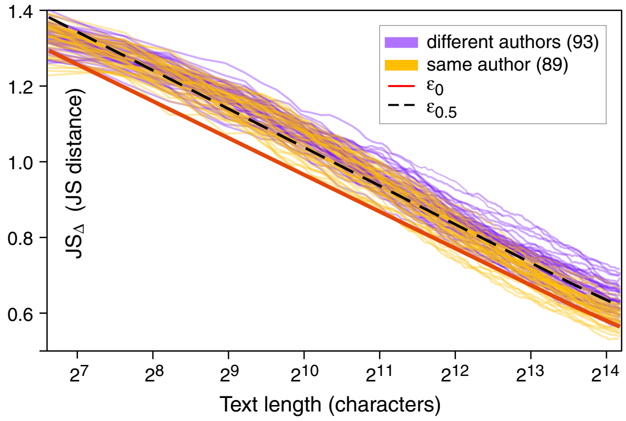


Figure 2: JS$_\Delta$ on the Webis Authorship Verification Corpus 2019 over text length. Each line corresponds to a text pair. The straight lines indicate the 0th and the 50th percentiles of distances within the true *different-authors* cases.

Kullback-Leibler divergence is defined as follows:

$$\mathrm{KLD}(P \| Q) = \sum_i P[i] \log \frac{P[i]}{Q[i]} \ , \tag{1}$$

where P and Q are discrete probability distributions corresponding to the relative frequencies of character trigrams in the to-be-obfuscated text and the known texts respectively. For true probability distributions, the KLD is always non-negative.

The KLD has shortcomings. First, it is asymmetric, so it is not entirely clear which character distribution should be P and which should be Q when comparing texts. Secondly, the KLD is defined only for distributions P and Q where $Q[i] = 0$ implies $P[i] = 0$. Conversely, $P[i] = 0$ yields a zero summand. Since we want to avoid reducing or skewing the measure further by "subsetting" or smoothing the trigrams, we resort to the Jensen-Shannon distance JS$_\Delta$ [14] in lieu of the KLD. The JS$_\Delta$ is a metric based on the symmetric Jensen-Shannon divergence (JSD) that is defined as

$$\mathrm{JSD}(P \| Q) = \frac{\mathrm{KLD}(P \| M) + \mathrm{KLD}(Q \| M)}{2} \ , \tag{2}$$

with

$$M = \frac{P + Q}{2} \ . \tag{3}$$

Introducing the artificial distribution M circumvents the KLD's problem of samples of one distribution being unknown in the other. Since $M[i]$ can never be 0 for any i with $P[i] + Q[i] > 0$, all summands of either $\mathrm{KLD}(P \| M)$ or $\mathrm{KLD}(Q \| M)$ must also be non-zero. Using the base-2 logarithm in the KLD, the JSD is $[0, 1]$-bounded. The JS$_\Delta$ metric is derived as

$$\mathrm{JS}_\Delta(P, Q) = \sqrt{2 \cdot \mathrm{JSD}(P \| Q)} \ . \tag{4}$$

Threshold	Obfuscation level	Slope	Intercept
$< \varepsilon_0$	No Obfuscation	n / a	n / a
$\geq \varepsilon_0$	Moderate Obfuscation	−0.099	1.936
$\geq \varepsilon_{0.5}$	Strong Obfuscation	−0.103	2.056
$\geq \varepsilon_{0.7}$	Stronger Obfuscation	−0.104	2.083
$> \varepsilon_{0.99}$	Over-obfuscation	−0.107	2.168

Table 2: Obfuscation levels and their log-scale polynomial fit coefficients on our training corpus.

4.2 Length-dependent Obfuscation

Employing a fixed JS$_\Delta$ threshold as the obfuscation target is a bad idea: it leads to over- or under-obfuscation for text pairs that have an a-priori high or low style distance. We also noted that JS$_\Delta$ is inversely correlated with text length: pairs of long texts are less distant to one another than pairs of short texts, since the shorter a text the sparser and noisier is its trigram distribution. This even holds if the texts are written by the same author. Figure 2 plots the JS$_\Delta$ over text length in our training data, revealing an approximately logarithmic relationship. The most interesting observation is the almost length-invariant spread of the resulting curves. Moreover, depending on their class, the curves converge towards the upper or lower bounds of this spread with growing length, thus becoming visibly separated.

Assuming that the observed JS$_\Delta$-to-length relationship generalizes to other text pairs of similar length—a hypothesis which merits further investigation in future work—, we measure style distance in JS$_\Delta$@L (Jensen-Shannon distance at length) and fit threshold lines to define obfuscation levels. Table 2 details the obfuscation levels ε_k corresponding to a linear least-squares fit on the logarithmic scale through a given level's k-th percentile of the distribution of JS$_\Delta$ in the *different-authors* class; the 0th percentile ε_0 and the 50th percentile $\varepsilon_{0.5}$ are displayed in Figure 2. The ε_0 threshold serves as an obfuscation baseline, indicating a *same-author* case as unobfuscated, if the JS$_\Delta$ between its documents is below this threshold. Otherwise, we call the obfuscation moderate, strong, stronger, and over-obfuscated, depending on the threshold the JS$_\Delta$ exceeds. We perform our further experiments with the $\varepsilon_{0.7}$ threshold, unless stated otherwise.

Regarding the line fit coefficients given in Table 2, the gradients of higher ε thresholds are slightly steeper, providing further evidence of the convergence rate of *same-author* cases. The ε_0 threshold line will cross the x axis for text lengths of $x \approx 2^{19.5}$ characters. Since negative distances are not sensible, such book-sized texts may be split into smaller chunks, which then can be obfuscated individually. Note that we were able to reproduce these threshold observations on the PAN 2014 novels corpus [48], albeit obtaining slightly different coefficients. In practice, we recommend training the coefficients on an appropriate corpus matching genre and register of the to-be-obfuscated texts.

4.3 Ranking Trigrams for Obfuscation

Key idea to yield a strong obfuscation is to iteratively change the frequency of those trigrams of the to-be-obfuscated text for which the positive impact on JS_Δ is maximal. In each iteration we hence rank the trigrams by their influence on JS_Δ via their partial KLD derivative, assuming that probability distribution Q represents the text that is to be obfuscated:

$$\frac{\partial}{\partial Q[i]}\left(P[i]\log_2\frac{P[i]}{Q[i]}\right)=-\frac{P[i]}{Q[i]\ln 2}\ . \tag{5}$$

Omitting constants, we get the rank-equivalent

$$\mathrm{R}_{\mathrm{KL}}(i)=\frac{P[i]}{Q[i]}\ . \tag{6}$$

R_{KL} gets larger with smaller $Q[i]$. I.e., a single obfuscation step boils down to *removing* one occurrence of the most influential trigram from the to-be-obfuscated text. This can be done naively by simply "cutting it out", or, more sensibly, via a targeted paraphrasing operation replacing a text passage with the trigram by another semantically equivalent text passage without the trigram. Then, the trigrams are re-ranked and the procedure is repeated until the JS_Δ exceeds the desired obfuscation threshold. We call this strategy *obfuscation by reduction.* Reversing the roles of P and Q yields an *addition* strategy, which we leave for future work.

Though the described greedy obfuscation effectively hinders verification, the naive cut-it-out approach results in rather unreadable texts and may even be easily "reverse engineered" by an informed verifier. Even with more sophisticated paraphrasing operations, a reverse-engineering attack against the greedy strategy seems plausible. We will address both shortcomings by an informed search, which is introduced in the next section.

5 Obfuscation via Informed Search

An author of a to-be-obfuscated text does obviously not wish their text to be "foozled" due to obfuscation (e.g., by naively cutting out trigrams). The text has to convey the same message as before and, ideally, it should look "inconspicuous" to an extent that readers do not suspect tampering [40]. However, automatic paraphrasing is still in its infancy: Beyond synonym substitution, paraphrasing operators targeting single words have only rarely been devised so far [16, 17, 55]. Still, the paraphrasing operators we are looking for do not have to alter a text substantially, which enables us to better estimate an operator's negative impact on text quality. Furthermore, similar to the presented greedy obfuscation, we can stop modifying a text when the desired obfuscation threshold is reached, which renders our approach "minimally invasive." The optimization goals can be summarized as follows:

1. Maximize the obfuscation as per the JS_Δ beyond a given ε_k without "over-obfuscating."
2. Minimize the accumulated text quality loss from consecutive paraphrasing operations.
3. Minimize the number of text operations.

We describe the problem as a potentially infinite space of possible (text) states, in which each state is reachable from one or multiple nodes in a graph spanned over the entire space by operators with accruing costs that transition from one state to another. At each node it is to be decided in which order to explore successor states so as to find a minimum-cost path from the starting node to a node that satisfies a pre-determined goal condition (i.e., sufficient obfuscation). Informed heuristic search is our choice to tackle this hard optimization problem. We will analyze also the admissibility property in order (1) to understand (in terms of modeling) the nature of the problem, and (2) to be able to compute an optimum solution if time and space constraints permit. However, due to the exponential size of the induced state space (text versions as nodes, paraphrasing operators as edges), one may give up admissibility while staying within acceptable error bounds. In the following, we will derive an admissible obfuscation heuristic and suggest a small, viable set of basic paraphrasing operators as an initial proof of concept.

5.1 An Admissible Obfuscation Heuristic

Let $h(n)$ denote a heuristic estimating the optimal cost for reaching a desired obfuscation threshold from node n, and let $g(n)$ denote the path costs to n starting at the original text node s.

Applying a paraphrasing operator has a highly non-linear effect on text quality (some changes are inconspicuous, others are not) and may also restrict the set of applicable operators (in the same text). For instance, applying the same operator a third time in a row may entail higher (quality) costs compared to applying it for the first time. This means that different paths from s to n can come with different estimations for the rest cost $h(n)$—in a nutshell, the parent discarding property may not hold [38]. A similar effect, but rooted in a different cause, results from the observation that some authors' texts are easier to obfuscate than others. We can address both issues and reinstall the conditions for parent discarding and admissible search by updating the operator costs for future application beyond node n, such that $g(n)$ turns into "normalized path costs."

Based on both the desired obfuscation threshold ε and the JS distance $\mathrm{JS}_{\Delta n}$ of the text at node n to the other text(s) from the same author, we define the prior heuristic as

$$h_{prior}(n)=\varepsilon-\mathrm{JS}_{\Delta n}. \tag{7}$$

The normalized path costs g_{norm} are defined as the cost-to-gain ratio of the accumulated path costs $g(n)$ to total JS_Δ change from start node s:

$$g_{norm}(n) = \frac{g(n)}{\mathrm{JS}_{\Delta n} - \mathrm{JS}_{\Delta s}}. \tag{8}$$

Finally, the heuristic $h(n)$ is defined as the product of $h_{prior}(n)$ and $g_{norm}(n)$:

$$h(n) = (\varepsilon - \mathrm{JS}_{\Delta n}) \cdot \frac{g(n)}{\mathrm{JS}_{\Delta n} - \mathrm{JS}_{\Delta s}}. \tag{9}$$

The prior heuristic guarantees convergence towards zero as we approach a goal node that exceeds the obfuscation threshold ε, while the normalized path costs determine the slope of the heuristic.

Consistency and Admissibility. A heuristic $h(n)$ is admissible if it does not exceed $h^*(n)$, the true cost of reaching an optimum goal via state n, for all n in the search space. Monotonicity $h(n) \leq c(n, n') + h(n')$ is a sufficient condition for admissibility yet easier to show. Rewriting it as

$$-h(n') + h(n) \leq g(n') - g(n),$$

and inserting in the heuristic Equation 9 yields

$$-\frac{(\varepsilon - \mathrm{JS}_{\Delta n'}) \cdot g(n')}{\mathrm{JS}_{\Delta n'} - \mathrm{JS}_{\Delta s}} + \frac{(\varepsilon - \mathrm{JS}_{\Delta n}) \cdot g(n)}{\mathrm{JS}_{\Delta n} - \mathrm{JS}_{\Delta s}} \leq g(n') - g(n)\ .$$

Defining $\bar{g}(n) = \mathrm{JS}_{\Delta n} - \mathrm{JS}_{\Delta s}$ as change function and inserting previous definitions we get

$$\frac{-h_{prior}(n') \cdot g(n')}{\bar{g}(n')} - \frac{-h_{prior}(n) \cdot g(n)}{\bar{g}(n)} \leq g(n') - g(n)\ .$$

We know $h_{prior}(n)$ to be monotonically decreasing, inverse to $\bar{g}(n)$, and converging towards zero as we approach a goal. If the cost and change functions $g(n)$ and $\bar{g}(n)$ are equivalent up to scale, they cancel each other out (up to scale), the slope of their quotient becomes zero, and the inequality turns into equality. Otherwise, if $g(n)$ dominates $\bar{g}(n)$, the inequality still holds. Though, if $\bar{g}(n)$ dominates $g(n)$, the sign of the quotient's gradient flips (as can be proved by the quotient rule), breaking the inequality and violating consistency. But since JS_Δ is bounded by $\sqrt{2}$ globally, the change function $\bar{g}(n)$ cannot be superlinear.

Limitations of our argument: (1) occasionally $\bar{g}(n)$ can locally dominate $g(n)$, and (2) both functions are presumed differentiable at n. In practice, the latter may hardly ever be true as texts are noisy, text operation side effects are unpredictable, and, the cumulative change function is not guaranteed to be monotonic. Still, step costs $c(n, n')$ will never be negative, which makes $g(n)$ monotonic but not necessarily differentiable. Thus, the heuristic function will not be fully consistent and may even overestimate.

	Operator name	**Cost value**
(1)	n-gram removal	40
(2)	Character flips	30
	Context-free synonyms	10
	Context-free hypernyms	6
	Context-dependent replacement	4
	Character maps	3
	Context-dependent deletion	2

Table 3: Implemented text operators and their assigned step costs in our heuristic obfuscation prototype.

In a practical scenario we can directly control the cost but not the change function, so we will have to deal with problems of overestimation and local optima. Generally, the first few steps of a search path are the most problematic since with little prior information the heuristic has to extrapolate based on very few data points, but is still expected to accurately estimate the remaining costs. Hence, an early heuristic is particularly susceptible to noise and can only give a coarse estimate. With more cumulative cost and change information available, the heuristic will stabilize towards the mean cost-gain proportion and eventually converge. This stabilization occurs quickly. In real application scenarios, we keep overestimation at a minimum or even avoid it at all and therefore obtain an approximately admissible heuristic due to the JS_Δ's boundedness.

5.2 Search Space Challenges

Given a longer text (one page or more), the number of potential operator applications is high. The most direct way to expand a node is to generate a successor with each applicable operator for each occurrence of each selected n-gram, but this will inevitably result in an immense number of very similar states with identical costs and almost identical JS_Δ change. I.e., the main challenge is to find a sensible middle ground between accepting a non-optimal solution too quickly or not finding a solution at all. Recall that one can easily turn the A* search into a depth-first or breadth-first search by making successor generation too cheap or too costly: depth-first search will always find a (non-optimal) solution after a sufficient number of operations, while breadth-first will never terminate before running out of memory.

We can accept a near-optimal solution, so selecting one or two occurrences of an n-gram (instead of all) will be sufficient. A potential problem is that the applicability of a high-quality operator is often restricted. However, one can increase the application probability by selecting not only the top-ranked n-gram but a small number of different near-top n-grams. This way, we have multiple high-impact n-grams with different contexts to work with, and we increase the chances of applying the operator opening alternative paths for the search. In practice, JS_Δ change is not a monotonic function

Efficiency	**Cases**		**Median**		
	Subset	#	Greedy	A*	Gain
Total operations	all	41	148	145	−2 %
	1+ ops	28	241	202	−16 %
	100+ ops	21	291	236	−19 %
Path costs	all	41	5,960	1,968	−67 %
	1+ ops	28	9,680	2,712	−72 %
	100+ ops	21	11,680	2,935	−75 %

Table 4: Efficiency of greedy obfuscation vs heuristic obfuscation for an obfuscation threshold of $\varepsilon_{0.5}$.

and steepest-ascent hill climbing does not guarantee an overall lowest-cost path. Thus, we applied each operator to two occurrences of the top ten n-grams and selected from these (up to 140 successors) six randomly for expansion. However, even with only six successors we still generate millions of nodes very quickly and will eventually run out of memory without finding a solution. Fortunately, we can assume that exploring more neighbors will not produce much better results after a while, so we can restart the search from a few promising lowest-cost nodes and discard others.

5.3 Paraphrasing Operators

Our prototype employs the seven basic text operators shown in Table 3 with costs assigned by us according to our appraisal of their negative impact on text quality. These are to be understood as a pilot study, more state-of-the-art text generation operators can be added easily. The most versatile yet most disruptive basic modification are (1) the removal of an n-gram, and (2) flipping two of its (or adjacent) characters. Such operations only are a last resort, and we hence set their costs much higher than those of other operators. As steps towards real paraphrasing, we also perform context-free synonym and hypernym replacement based on WordNet [36] as well as context-dependent replacements and deletions using the word 5-gram model of Netspeak [52]. Lastly, we created a map of similar punctuation characters for inconspicuous character swaps.

6 Evaluation

To evaluate our approach, we report on: (1) an efficiency comparison of greedy versus heuristic obfuscation, (2) an effectiveness analysis against well-known authorship verification approaches (unmasking, compression-based models, and PAN participants), as well as (3) a review and discussion of an example obfuscated text.

Our experiments are based on PAN authorship corpora and the Webis Authorship Verification Corpus 2019, half of them *same-author* cases, the other half *different-authors* cases (each a pair of texts of about 23,000 characters / 4,000 words). Instead of the more particular genres studied at PAN, our new corpus contains longer texts and more modern literature from Project Gutenberg. We also took extra care to cleanse the plain text, unified special characters, and removed artifacts; in particular, we ensured that no author appears in more than one case. The training-test split is 70-30 so as to have a decent training portion. The corpus is released alongside the code of our framework and other data.

Confidence	**Unobfuscated**			**Obfuscated**		
Hyperplane	Classified	Effectiveness		Classified	Effectiveness	
threshold	cases [%]	Prec.	Rec.	cases [%]	Prec.	Rec.
0.8	11.3	1.00	0.17	2.5	1.00	0.02
0.7	15.0	1.00	0.24	6.2	1.00	0.05
0.6	18.8	1.00	0.24	11.3	0.75	0.07
0.5	26.3	1.00	0.29	24.0	0.86	0.15
0.0	100.0	0.74	0.63	100.0	0.71	0.42

Table 5: Unmasking performance on our test data at various confidence thresholds before and after obfuscation. Recall treats unclassified cases as false negatives.

6.1 Search Over Greedy Obfuscation

Table 4 contrasts the efficiency of the greedy obfuscation with that of our heuristic search approach, measured in terms of medians of total text operations and path costs. Heuristic search achieves a decrease of operations of up to 19% for texts that need at least 100 operations and an accumulated path cost decrease of up to 75%. Since the greedy obfuscation approach cannot choose among different operators, it must rely on the most effective one to achieve the obfuscation goal, incurring significant path costs. Given that both obfuscators employ adaptive thresholds, there are cases which do not require any (or only little) obfuscation, whereas others need more than 100. The latter are of particular interest, since it is here where heuristic obfuscation outperforms greedy obfuscation the most.

6.2 Obfuscation Against Unmasking

As our first and primary evaluation, we test the heuristic obfuscation approach against the generalized unmasking approach presented in Section 3. Due to the way unmasking successively degrades the text representations in-depth, fooling unmasking verification provides us with evidence that our obfuscation technique works at a deeper level than just the few most superficial text features. Another very neat property of unmasking is that it produces visual curve representations of the process, which render the effects of obfuscation accessible to human inspection and interpretation and therefore make unmasking a very suitable benchmark.

As described previously, we use the chunk frequencies of the 250 most common words as features, determine classification accuracy by 10-fold cross validation using an SVM classifier, and remove ten features per iteration. Table 5 contrasts the performance of unmasking before and after obfuscation. With increasing confidence

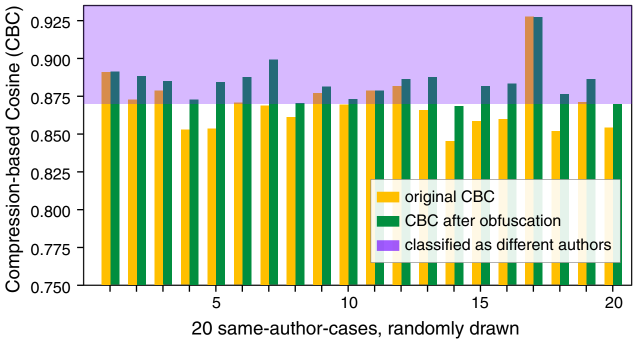


Figure 3: CBC values of 20 PPMD-compressed *same-author* pairs before and after obfuscation up to the obfuscation threshold $\varepsilon_{0.7}$. The classification threshold by which *same-author* and *different-authors* cases could be distinguished is highlighted in the top portion.

thresholds, between 19 % down to 11 % of the cases are decidable before obfuscation, decreased by a factor of 2 to 4 after obfuscation. On average, 205 trigrams were obfuscated; as little as about 3 % of a text.

6.3 Obfuscation Against Compression Models

Another verification approach that differs from traditional feature-engineering are compression-based models. We use the approach by Halvani et al. [19], who recommend the compression-based cosine (CBC) by Sculley and Brodley [45] calculated on the text pairs after compression with the PPMD algorithm [21].

Figure 3 shows CBC values on a random selection of 20 exemplary *same-author* cases from our test dataset before and after obfuscation with the decision threshold highlighted. Quite impressively, almost none of the cases are classified correctly anymore after obfuscation. Overall, the accuracy drops from originally 78 % to 60 % in the case of CLM and in the case of CBC, respectively, from 71 % to 55 %, which is equivalent to random guessing. This strong effect can be explained as follows: Sculley and Brodley describe their metrics in terms of the Kolmogorov complexity, but the reason why natural language allows for very good compression ratios is its predictability (printed English has an entropy of at most 1.75 bits per character [11]). PPMD uses finite-order Markov language models for compression, which are effective at predicting characters in a sentence, but sensitive to the increased entropy that stems as an immediate result from our obfuscation.

6.4 PAN Obfuscation Evaluation

We further conducted an extensive evaluation of our obfuscation scheme against the top submissions to the verification task at PAN 2013–2015 [24, 48, 50]. The results are shown in Table 7. On all verifiers tested, we achieve an average AUC and C@1 reduction of around 10 and 6 percentage points on three of the corpora. With only minimal text modifications, this puts us in second place on the PAN13 and PAN15 corpora, and fourth on PAN14 Essays compared to other obfuscators submitted to PAN [18] (Table 6). The PAN14 Novels corpus turns out to be the most challenging for our approach and there are multiple reasons for that. First, the texts are significantly longer. This makes it difficult to assess the overall obfuscation with a global measure like JS_Δ. As a result, only few sentences were actually obfuscated with most of the text left untouched. Insofar, we were surprised to see any significant effect at all (best individual result: 13 percentage points). To make matters worse, the flat search landscape spanned by our obfuscation operators leads to an increasing number of reopened states on these longer texts, greatly reducing the efficiency of the heuristic search. This reveals an important detail to explore in future work: obfuscation operations need to be distributed across the whole text and progress needs to be measured on smaller parts of it to ensure uniform obfuscation of everything and avoid obfuscation "hot spots". Secondly, the number of "known" texts varies substantially, which demands more research into how we can calculate a minimal yet sufficient JS_Δ@L stopping criterion if a larger amount of known material is available. Thirdly, the corpus consists primarily of works by H. P. Lovecraft paired with fan fiction, which incurs unforeseeable global corpus features that verifiers can exploit, but which we do not consider for obfuscation. Lastly, we identify *kocher15* [30] as the most difficult verifier for us to obfuscate. Employing an impostor approach on the most frequent words, it was not the best-performing verifier in the first place, but proves most resilient against our "reductive" obfuscation, which tends to obfuscate only n-grams that are already rare for maximum effect. We expect that augmenting a reduction obfuscation with the previously-mentioned extension strategy will yield better results and an overall safer obfuscation.

Obfuscator	**PAN13**		**PAN14 E**		**PAN14 N**		**PAN15**	
	Δ AUC / C@1							
our approach	0.10	0.06	0.09	0.05	0.03	0.02	0.11	0.06
mihaylova	0.11	0.08	0.13	0.11	0.16	0.11	0.11	0.09
castro	0.11	0.05	0.13	0.11	0.13	0.08	0.09	0.06
keswani	0.09	0.07	0.11	0.09	0.10	0.07	0.06	0.05
bakhteev	0.05	0.05	0.05	0.05	0.07	0.05	0.06	0.06
mansoorizadeh	0.04	0.03	0.05	0.05	0.06	0.04	0.04	0.03

Table 6: Obfuscation performance comparison for C@1 and ROC-AUC against PAN participants.

6.5 Example of an Obfuscated Text

Assessing the text quality in tasks that involve generation, such as translation, paraphrasing, and summarization, is still mostly manual work. Frequently used measures like ROUGE cannot be applied in the context of obfuscation, since our obfuscated texts are up to 97 % identical to their unobfuscated versions. This is why

Verifier	Unobfuscated			Obfuscated			Difference		
	AUC	C@1	FS	AUC	C@1	FS	AUC	C@1	FS
a) PAN13									
bagnall15	0.86	0.79	0.68	0.74	0.64	0.48	0.11	0.15	0.20
castillojuarez14	0.49	0.43	0.21	0.50	0.53	0.27	-0.02	-0.10	-0.06
castro15	0.93	0.77	0.71	0.87	0.73	0.64	0.06	0.03	0.08
frery14	0.62	0.57	0.35	0.37	0.40	0.15	**0.25**	0.17	0.20
khonji14	0.86	0.76	0.65	0.70	0.60	0.42	0.16	0.16	0.23
kocher15	0.75	0.64	0.48	0.77	0.65	0.50	-0.02	-0.01	-0.02
layton14	0.62	0.67	0.41	0.47	0.53	0.25	0.15	0.13	0.16
mezaruiz14	0.75	0.65	0.49	0.57	0.53	0.30	0.18	0.12	0.19
mezaruiz15	0.73	0.71	0.52	0.50	0.53	0.26	0.24	**0.18**	**0.26**
modaresi14	0.50	0.50	0.25	0.47	0.50	0.24	0.03	0.00	0.02
moreau14	0.77	0.62	0.48	0.61	0.51	0.32	0.16	0.11	0.17
moreau15	0.71	0.47	0.33	0.60	0.47	0.28	0.12	0.00	0.05
singh14	0.39	0.33	0.13	0.44	0.43	0.19	-0.06	-0.10	-0.06
zamani14	0.75	0.70	0.53	0.71	0.70	0.50	0.05	0.00	0.03
Average							**0.10**	**0.06**	**0.10**
b) PAN14 Essays									
bagnall15	0.57	0.55	0.31	0.43	0.45	0.19	0.14	0.10	0.12
castillojuarez14	0.55	0.58	0.32	0.55	0.58	0.32	0.00	0.00	0.00
castro15	0.62	0.59	0.36	0.51	0.53	0.27	0.11	0.05	0.09
frery14	0.72	0.71	0.51	0.68	0.68	0.46	0.04	0.03	0.05
khonji14	0.60	0.58	0.35	0.41	0.50	0.20	0.19	0.09	0.15
kocher15	0.63	0.59	0.37	0.61	0.57	0.35	0.02	0.02	0.02
layton14	0.59	0.61	0.36	0.51	0.53	0.27	0.08	0.08	0.09
mezaruiz14	0.57	0.56	0.32	0.49	0.51	0.25	0.08	0.04	0.07
mezaruiz15	0.52	0.52	0.27	0.32	0.37	0.12	**0.21**	**0.16**	**0.16**
modaresi14	0.60	0.58	0.35	0.57	0.57	0.32	0.04	0.01	0.03
moreau14	0.62	0.60	0.37	0.51	0.53	0.27	0.11	0.07	0.10
moreau15	0.57	0.52	0.30	0.50	0.51	0.26	0.07	0.01	0.04
singh14	0.70	0.66	0.46	0.61	0.61	0.37	0.09	0.04	0.08
zamani14	0.58	0.55	0.32	0.48	0.49	0.23	0.11	0.06	0.09
Average							**0.09**	**0.05**	**0.08**
c) PAN14 Novels									
bagnall15	0.68	0.68	0.47	0.61	0.59	0.36	0.07	0.09	0.10
castillojuarez14	0.63	0.62	0.39	0.59	0.56	0.33	0.04	0.05	0.06
castro15	0.64	0.51	0.33	0.50	0.39	0.19	**0.14**	**0.12**	**0.13**
frery14	0.61	0.59	0.36	0.59	0.57	0.34	0.02	0.02	0.02
khonji14	0.75	0.61	0.46	0.71	0.58	0.41	0.04	0.03	0.05
kocher15	0.63	0.57	0.36	0.66	0.59	0.39	-0.03	-0.02	-0.03
layton14	0.51	0.51	0.26	0.50	0.50	0.25	0.01	0.01	0.01
mezaruiz14	0.66	0.61	0.41	0.64	0.62	0.40	0.02	0.00	0.01
mezaruiz15	0.56	0.51	0.28	0.57	0.51	0.29	-0.01	0.00	0.00
modaresi14	0.71	0.72	0.51	0.69	0.69	0.47	0.02	0.03	0.03
moreau14	0.60	0.52	0.31	0.56	0.51	0.29	0.04	0.01	0.03
moreau15	0.64	0.50	0.32	0.61	0.53	0.32	0.03	-0.03	0.00
singh14	0.66	0.58	0.38	0.63	0.56	0.35	0.03	0.02	0.03
zamani14	0.73	0.65	0.48	0.71	0.63	0.44	0.03	0.02	0.03
Average							**0.03**	**0.02**	**0.03**
d) PAN15									
bagnall15	0.81	0.76	0.61	0.72	0.71	0.51	0.09	0.05	0.10
castillojuarez14	0.64	0.64	0.41	0.55	0.55	0.30	0.09	0.09	0.11
castro15	0.75	0.69	0.52	0.72	0.68	0.49	0.03	0.01	0.03
frery14	0.54	0.46	0.25	0.47	0.43	0.20	0.07	0.04	0.05
khonji14	0.82	0.65	0.53	0.59	0.49	0.49	**0.23**	**0.16**	**0.24**
kocher15	0.74	0.69	0.51	0.72	0.66	0.48	0.02	0.02	0.03
layton14	0.67	0.50	0.34	0.49	0.50	0.25	0.18	0.00	0.09
mezaruiz14	0.65	0.61	0.40	0.55	0.54	0.30	0.10	0.07	0.10
mezaruiz15	0.74	0.69	0.51	0.55	0.53	0.29	0.19	**0.16**	0.22
modaresi14	0.40	0.41	0.16	0.39	0.40	0.16	0.01	0.00	0.00
moreau14	0.66	0.58	0.38	0.52	0.49	0.25	0.14	0.09	0.13
moreau15	0.71	0.64	0.45	0.52	0.49	0.26	0.19	0.15	0.20
singh14	0.78	0.50	0.39	0.66	0.50	0.33	0.12	0.00	0.06
zamani14	0.74	0.67	0.50	0.71	0.66	0.47	0.04	0.00	0.03
Average							**0.11**	**0.06**	**0.10**

Table 7: Results of the top verifiers of PAN 2013–2015 before and after obfuscating the four task corpora. FS (Final Score) is the product of ROC-AUC and C@1. On average, we degrade AUC by at least 10 and C@1 by about 6 percentage points on three of the corpora, though much less on the PAN14 Novels corpus. Most noticeably, we can reduce the FS of *bagnall15* [2] (winning submission of PAN 2015) by 10–20 percentage points on all four corpora. The best obfuscation results on each corpus are marked bold. Verifiers that were improved are highlighted in red.

we resort to manually inspecting obfuscated texts and the changes made. Below is an excerpt of an original text along with the obfuscations applied to it. Selected trigrams are underlined, removed words are struck out, and inserted words are highlighted:

> 'It was the only chance ~~we had~~w ehad to win.' Duke
> swallowed the idea slowly. He couldn't picture a
> ~~planet~~satellite giving up its last protection
> for ~~a~~phi desperate effort to end the war on purely
> offensive drive. Three billion people watching the
> home fleet take off, ~~knowing~~deciding the skies were
> ~~open~~resort for all the ~~hell~~mischief that a savage
> enemy could send! On Earth, the World Senate hadn't
> permitted the building of one ~~battleship~~frigate,
> for fear of reprisal. [...]

Excerpt of *Victory* by Lester del Rey

We selected an example where, by chance, different operators were applied in close vicinity. This "density" of operations is not representative. We can see both high- and low-quality replacements at work. Most can be attributed to the WordNet synonym operator. The replacement of "a" with "phi" is clearly such a case. The more suitable replacements originate from more context-dependent replacements, whereas "we had" → "w ehad" is a result of the flip operator.

For comparison with related work, we carried out a human assessment of a few random obfuscation samples as per the PAN obfuscation task. We achieved an overall grade of about 2.6 (1 = excellent, 5 = fail), which places us somewhere within the top three submissions.

While the obfuscated text probably is not fit for publication, it does look promising even with our basic set of paraphrasing operators. The text was generated within a few minutes and passes the verifiers without being recognized as a same-author case. Texts from other cases look similar: a mixture of poor and good operations, where according to our own review about half of the changes made are still rather nonsensical. Since our set of operators is just a proof of concept, we will devise more sophisticated ones and better weighting schemes in future work, which is vital for achieving acceptable text quality. Promising approaches already exist, such as neural editing, paraphrasing, and targeted neural style transfer [16, 17, 55].

7 Author Obfuscation Robustness

The strong effectiveness against existing authorship verification systems merits further analysis of the safety of an obfuscation against targeted reverse obfuscation attacks. In this regard, it is important to keep in mind that obfuscation safety against existing verification approaches does not necessarily imply safety against human readers, or targeted attackers. Although it is obviously desirable to obfuscate against all three, achieving only one goal may be sufficient in many cases. For example, a forensic linguist may never even start to investigate a text if it was not flagged by a broader general-purpose verification system before. Nevertheless, in what follows we devise a targeted attack to reverse the effect our heuristic obfuscation has, exploiting the fact that an attacker has perfect knowledge of how our algorithm works (i.e., Kerckhoff's principle), followed by an evaluation of its success.

7.1 Reverse Obfuscation Attack

Assuming the applied obfuscation operators were powerful enough to produce high-quality text able to go unnoticed by human readers (an assumption which is out of scope of our prototype, but one which we make with regard to recent advances in text generation), the most probable attack is one considering the text's inherent statistics. Algorithmic modification of a text's n-gram distribution inevitably leads to statistical anomalies if one is not particularly careful about maintaining an unsuspicious target distribution. The main shortcoming of maximum-effectiveness KLD obfuscation is its resulting skew towards unnaturally rare n-grams, which a statistical evaluation of the text could reveal. This in itself only gives away that the text was tampered with, but not necessarily its original author. At this point, however, we may fall victim to our basic experimental setup, which only considers two specific texts without additional context or general stylistic knowledge about their author. In this reduced scenario, a much more severe vulnerability arises if an attacker gets hold of the text that was used as a reference for sufficient obfuscation.

Our obfuscation approach manipulates the text's n-gram distribution with regard to maximizing the KLD gradient quotient $P[i]/Q[i]$ by reducing frequencies in Q. For a reverse obfuscation, the same approach can be followed through in the opposite direction, i.e., by repeatedly selecting the n-gram i which maximizes $P[i]/Q[i]$ and increasing its frequency in Q (thus minimizing the KLD). Although this reverse obfuscation will not restore the original text, it will gradually approximate its original distribution. The average speed at which the two text representations converge will be highest between the obfuscated text and the text that was used to perform the obfuscation. Moreover, an attacker is at a significant advantage, since they do not have to actually generate text but can simply adjust the n-gram writing style representations directly.

7.2 Evaluation

To show this effect in practice, we set up an experiment measuring the effect of reverse obfuscation on an obfuscated text against (1) the original obfuscation text, (2) another unseen text written by the same author, and (3) a random text by a different author. Since only measuring the influence of reverse obfuscation does not require actual text modifications (and we could not reliably restore the original n-grams at their original positions anyway), it is sufficient to show its effects on the abstract frequency vector representations. Figure 4 visualizes the meta classification hyperplane distances of the tested *same-author* text pairs before obfuscation, after obfuscation, and after reverse obfuscation, contrasting the use of the obfuscation text and an unrelated text side by side.

Using the Webis Authorship Verification Corpus, a naive greedy obfuscation with 100 iterations was able to reduce the unmasking meta classification (as described in Section 3) from 26 true positives to ten. A heuristic obfuscation using a strong adaptive threshold of $\varepsilon_{0.5}$ achieved a reduction to only eight true positives. After applying 40 iterations of reverse obfuscation with the original obfuscation text, however, we were able to increase this number to 28, which even exceeded our starting point of 26. The same observation held for a stronger obfuscation threshold of $\varepsilon_{0.7}$ where we saw an increase from zero true positives to eleven true positives after the same 40 iterations of reverse obfuscation. We repeated the experiment using a third text by the same author and finally a text by a different author. In either case, we could increase the number of true positives by only three. For restoring the original number of true positives, more than 200 iterations were necessary, obviously at the cost of equally many false positives.

7.3 Discussion

Although we were unable to find a distinction between the reverse obfuscation behavior of a third text by the same author and a text by a different author, we could clearly single out the text that was used originally for creating the obfuscation. Hence, these experiments provide us with crucial evidence that the obfuscation text assumes the role of a "key" which needs to be kept secret for the safety of the obfuscation. Unfortunately, such an attack is difficult to come by in terms of the experimental setup, since in practice only few text samples of an author are usually known. Over the 41 obfuscated texts in our test set, the heuristic obfuscation process created a total of 450 additional rare n-grams with a frequency of one. Using these rare n-grams as a starting point for reverse obfuscation quite obviously

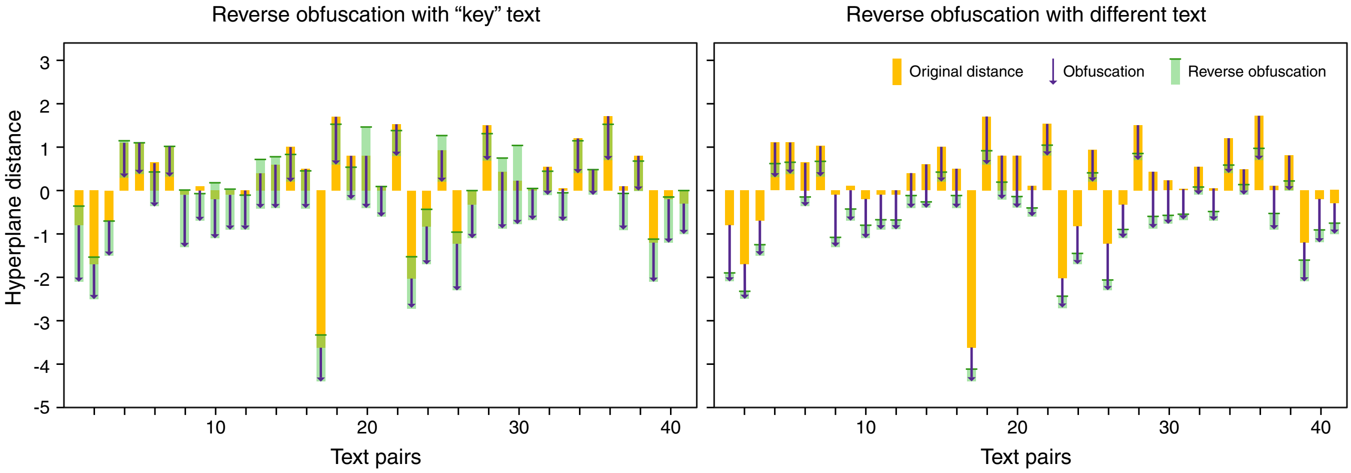


Figure 4: Distances of 41 same-author text pairs from the unmasking meta classification hyperplane before obfuscation (yellow), their relative shift downwards after a fixed 100 rounds of obfuscation (blue arrows), and finally their shift upwards after 40 iterations of reverse obfuscation (green). The left-hand graph shows reverse obfuscation using the original “key” text, the right-hand graph shows reverse obfuscation using a random text written by a different author. With the “key” text, a much faster approximation of the original distance can be seen for all pairs. In most of these cases, the reverse obfuscation even overshoots adding to the original hyperplane distance of the unobfuscated text.

results in the KLD gradient explosion we observed. As a possible countermeasure, less effective n-grams from the center of the distribution could be picked. This would result in more modifications to the text in order to achieve the same effectiveness, but it would allow for “hiding” of the manipulated n-grams between other n-grams of similar frequency. Such a more modest obfuscation could be augmented by a global optimization strategy for avoiding unwanted side effects, such as the inadvertent introduction of too many entirely new or unusual n-grams caused by word or character replacements, thus ensuring an overall unsuspicious target distribution. The latter could be achieved by measuring the divergence from a set of unrelated impostor texts or, more generally, the expected Zipf distribution. In theory, this adjusted approach may seem like an ideal obfuscation method, though it remains to be seen how effective this strategy can be applied in practice while also maintaining proper textual entailment with the original unobfuscated text.

8 Conclusion

We introduced a promising new paradigm for authorship obfuscation and implemented a first fully functional prototype that is able to obfuscate texts against an adapted variant of Koppel and Schler’s unmasking and other state-of-the-art approaches. We identified and addressed the following challenges: measuring style similarity in a manner that is agnostic to state-of-the-art verifiers, identifying those parts of a text that have the highest impact on style, and devising and analyzing a search heuristic amenable for informed search. Further, we identified potential vulnerabilities in the application of said approach and proposed remedies for making it more robust against reverse obfuscation attacks.

Our study opens up interesting avenues for future research, such as the development of more powerful, targeted paraphrasing operators and theoretical analyses of the search space properties. We consider heuristic search-based authorship obfuscation a key enabling technology that, combined with tailored deep generative models for paraphrasing, will yield better and stronger obfuscations.

M.Sc. Janek Bevendorff Janek Bevendorff graduated in Computer Science at the Bauhaus-Universität Weimar in 2018 and has since worked with the Webis group as a PhD candidate in the fields of natural language processing and big data analytics with focus on stylometry and authorship verification. In his Master’s thesis, he wrote about “Authorship Obfuscation Using Heuristic Search”, which part of the research presented in this paper is based on.

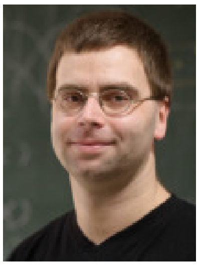
Address: Bauhaus-Universität Weimar, Germany

M.Sc. Tobias Wenzel Tobias Wenzel did his Master’s in Computer Science in 2019 at Leipzig University on the topic of authorship boosting for attacking KLD-based authorship obfuscation. His work established the ground work for the reverse obfuscation attacks discussed in this paper.

Address: Leipzig University, Germany

Jun.-Prof. Dr. Martin Potthast Martin Potthast is head of the Text Mining and Retrieval group at Leipzig University. His research areas include information retrieval and natural language processing, as well as applied machine learning, data mining, and crowdsourcing. Focus of his research is the development of algorithms and machine learning models for information systems and computational stylometry. Martin is co-initiator of the PAN network of excellence for the digital text forensic. Martin studied computer science at Paderborn University, obtained a PhD from the Bauhaus-Universität Weimar in 2011, where he also spent his Postdoc time at the Digital Bauhaus Lab, and was appointed Juniorprofessor at Leipzig University in 2017.

Address: Leipzig University, Germany



Prof. Dr. Matthias Hagen Matthias Hagen is Professor for “Big Data Analytics” at the Martin-Luther-Universität Halle-Wittenberg. His current research interests include information retrieval and web search (e.g., query understanding, conversational search), natural language processing (e.g., argumentation), and data analytics + mining (e.g., simulation and sensor data). Matthias studied computer science at the Friedrich-Schiller-Universität Jena where he also obtained his PhD on algorithmic and computational complexity issues of the equivalence test of monotone Boolean formulas. Afterwards, he moved to the Bauhaus-Universität Weimar where he lead the junior research group “Intelligentes Lernen” (intelligent learning) from 2008-2013. From 2013-2018, Matthias was Juniorprofessor for “Big Data Analytics” and lead the corresponding junior research group at the Bauhaus-Universität Weimar.

Address: Martin-Luther-Universität Halle-Wittenberg, Germany

Prof. Dr. Benno Stein Benno Stein is chair of the Web-Technology and Information Systems Group at the Bauhaus-Universität Weimar. His research focuses on modeling and solving data- and knowledge-intensive information processing tasks. Common ground of his research are the principles and methods of symbolic Artificial Intelligence. Benno has developed theories, algorithms, and tools for information retrieval, machine learning, natural language processing, knowledge processing, as well as for engineering design and simulation. He studied at Karlsruhe University (1984-1989), did his PhD (1995) and his habilitation (2002) in computer science at Paderborn University, and was appointed as a full professor for Web Technology and Information Systems at the Bauhaus-Universität Weimar (2005). He is cofounder and spokesperson of the Digital Bauhaus Lab, an interdisciplinary research center for Computer Science, Arts, and Engineering.

Address: Bauhaus-Universität Weimar, Germany

Literature

[1] A. Abbasi and H. Chen. Writeprints: A stylometric approach to identity-level identification and similarity detection in cyberspace. *ACM Trans. Inf. Syst.*, 26(2):7:1–7:29, Apr. 2008.

[2] D. Bagnall. Author Identification using multi-headed Recurrent Neural Networks—Notebook for PAN at CLEF 2015. In *CLEF 2015 Evaluation Labs and Workshop – Working Notes Papers*.

[3] J. Bevendorff, M. Potthast, M. Hagen, and B. Stein. Heuristic Authorship Obfuscation. In *57th Annual Meeting of the Association for Computational Linguistics (ACL 2019)*, pages 1098–1108. Association for Computational Linguistics, July 2019.

[4] J. Bevendorff, B. Stein, M. Hagen, and M. Potthast. Generalizing Unmasking for Short Texts. In *14th Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL 2019)*, pages 654–659. Association for Computational Linguistics, June 2019.

[5] J. Bevendorff, B. Stein, M. Hagen, and M. Potthast. Bias Analysis and Mitigation in the Evaluation of Authorship Verification. In *57th Annual Meeting of the Association for Computational Linguistics (ACL 2019)*, pages 6301–6306. Association for Computational Linguistics, July 2019.

[6] H. Bo, S. H. H. Ding, B. C. M. Fung, and F. Iqbal. ER-AE: differentially-private text generation for authorship anonymization. *CoRR*, abs/1907.08736, 2019.

[7] B. T. Boenninghoff, S. Hessler, D. Kolossa, and R. M. Nickel. Explainable authorship verification in social media via attention-based similarity learning. *CoRR*, abs/1910.08144, 2019.

[8] D. Boumber, Y. Zhang, M. Hosseinia, and A. Mukherjee. Robust Authorship Verification with Transfer Learning. 2019.

[9] E. G. Bourne. The authorship of the federalist. *The American Historical Review*, 2(3):443–460, 1897.

[10] M. Brennan, S. Afroz, and R. Greenstadt. Adversarial Stylometry: Circumventing Authorship Recognition to Preserve Privacy and Anonymity. *ACM Trans. Inf. Syst. Secur.*, 15(3):12, 2012.

[11] P. F. Brown, S. D. Pietra, V. J. D. Pietra, J. C. Lai, and R. L. Mercer. An estimate of an upper bound for the entropy of english. *Computational Linguistics*, 18(1):31–40, 1992.

[12] D. Castro, Y. Adame, M. Pelaez, and R. Muñoz. Authorship Verification, Combining Linguistic Features and Different Similarity Functions—Notebook for PAN at CLEF 2015. In *CLEF 2015 Evaluation Labs and Workshop – Working Notes Papers*.

[13] C. Emmery, E. M. Arévalo, and G. Chrupala. Style obfuscation by invariance. In *Proceedings of the 27th International Conference on Computational Linguistics, COLING 2018, Santa Fe, New Mexico, USA, August 20-26, 2018*, pages 984–996, 2018.

[14] D. M. Endres and J. E. Schindelin. A new metric for probability distributions. *IEEE Trans. Information Theory*, 49(7):1858–1860, 2003.

[15] J. Fréry, C. Largeron, and M. Juganaru-Mathieu. UJM at CLEF in Author Identification—Notebook for PAN at CLEF 2014. In In *CLEF 2014 Evaluation Labs and Workshop – Working Notes Papers.*

[16] D. Grangier and M. Auli. Quickedit: Editing text & translations via simple delete actions. *CoRR*, abs/1711.04805, 2017.

[17] K. Guu, T. B. Hashimoto, Y. Oren, and P. Liang. Generating sentences by editing prototypes. *CoRR*, abs/1709.08878, 2017.

[18] M. Hagen, M. Potthast, and B. Stein. Overview of the Author Obfuscation Task at PAN 2017: Safety Evaluation Revisited. In *Working Notes Papers of the CLEF 2017 Evaluation Labs*, volume 1866 of *CEUR Workshop Proceedings.*

[19] O. Halvani, C. Winter, and L. Graner. Authorship verification based on compression-models. *CoRR*, abs/1706.00516, 2017.

[20] O. Halvani, C. Winter, and L. Graner. Assessing the applicability of authorship verification methods. In *Proceedings of the 14th International Conference on Availability, Reliability and Security, ARES 2019, Canterbury, UK, August 26-29, 2019*, pages 38:1–38:10, 2019.

[21] P. G. Howard. The design and analysis of efficient lossless data compression systems. Brown University, 1993.

[22] F. Iqbal, R. Hadjidj, B. C. Fung, and M. Debbabi. A novel approach of mining write-prints for authorship attribution in e-mail forensics. *Digital Investigation*, 5:S42–S51, 2008.

[23] P. Juola. Authorship Attribution. *Foundations and Trends Information Retrieval*, 1(3):233–334, Dec. 2006.

[24] P. Juola and E. Stamatatos. Overview of the Author Identification Task at PAN 2013. In P. Forner, R. Navigli, and D. Tufis, editors, *CLEF 2013 Evaluation Labs and Workshop – Working Notes Papers, 23-26 September, Valencia, Spain.* CEUR-WS.org, Sept. 2013.

[25] P. Juola and D. Vescovi. Analyzing Stylometric Approaches to Author Obfuscation. In *Advances in Digital Forensics VII - 7th IFIP WG 11.9 International Conference on Digital Forensics, Orlando, FL, USA, January 31 - February 2, 2011, Revised Selected Papers*, volume 361 of *IFIP Advances in Information and Communication Technology*, pages 115–125.

[26] G. Kacmarcik and M. Gamon. Obfuscating Document Stylometry to Preserve Author Anonymity. In *ACL 2006, 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference, Sydney, Australia, 17-21 July 2006.*

[27] M. Kestemont, K. Luyckx, W. Daelemans, and T. Crombez. Cross-genre authorship verification using unmasking. *English Studies*, 93(3):340–356, 2012.

[28] D. V. Khmelev and W. J. Teahan. A repetition based measure for verification of text collections and for text categorization. In *SIGIR 2003: Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, July 28 - August 1, 2003, Toronto, Canada*, pages 104–110.

[29] M. Khonji and Y. Iraqi. A Slightly-modified GI-based Author-verifier with Lots of Features (ASGALF)—Notebook for PAN at CLEF 2014. In *CLEF 2014 Evaluation Labs and Workshop – Working Notes Papers.*

[30] M. Kocher and J. Savoy. UniNE at CLEF 2015: Author Identification—Notebook for PAN at CLEF 2015. In *CLEF 2015 Evaluation Labs and Workshop – Working Notes Papers.*

[31] M. Kocher and J. Savoy. A simple and efficient algorithm for authorship verification. *JASIST*, 68 (1):259–269, 2017.

[32] M. Kocher and J. Savoy. Distance measures in author profiling. *Inf. Process. Manage.*, 53(5): 1103–1119, 2017.

[33] M. Koppel and J. Schler. Authorship Verification as a One-Class Classification Problem. In C. Brodley, editor, *Proceedings of the Twenty-First International Conference on Machine Learning*, pages 1–7.

[34] A. Mahmood, F. Ahmad, Z. Shafiq, P. Srinivasan, and F. Zaffar. A girl has no name: Automated authorship obfuscation using mutant-x. *PoPETs*, 2019(4):54–71, 2019.

[35] A. McDonald, S. Afroz, A. Caliskan, A. Stolerman, and R. Greenstadt. Use Fewer Instances of the Letter "i": Toward Writing Style Anonymization. In S. Fischer-Hübner and M. Wright, editors, *Privacy Enhancing Technologies - 12th International Symposium, PETS 2012, Vigo, Spain, July 11-13, 2012. Proceedings*, volume 7384 of *Lecture Notes in Computer Science*, pages 299–318.

[36] G. A. Miller. Wordnet: A lexical database for english. *Commun. ACM*, 38(11):39–41, Nov. 1995.

[37] A. Narayanan, H. S. Paskov, N. Z. Gong, J. Bethencourt, E. Stefanov, E. C. R. Shin, and D. Song. On the feasibility of internet-scale author identification. In *IEEE Symposium on Security and Privacy, SP 2012, 21–23 May 2012, San Francisco, California, USA*, pages 300–314.

[38] J. Pearl. *Heuristics - intelligent search strategies for computer problem solving.* Addison-Wesley series in artificial intelligence.

[39] N. Potha and E. Stamatatos. Improved algorithms for extrinsic author verification. *Knowledge and Information Systems*, oct 2019.

[40] M. Potthast, M. Hagen, and B. Stein. Author Obfuscation: Attacking the State of the Art in Authorship Verification. In *Working Notes Papers of the CLEF 2016 Evaluation Labs*, volume 1609 of *CEUR Workshop Proceedings*.

[41] M. Potthast, F. Schremmer, M. Hagen, and B. Stein. Overview of the Author Obfuscation Task at PAN 2018: A New Approach to Measuring Safety. In L. Cappellato, N. Ferro, J.-Y. Nie, and L. Soulier, editors, *Working Notes Papers of the CLEF 2018 Evaluation Labs*, volume 2125 of *CEUR Workshop Proceedings*.

[42] J. Rao and P. Rohatgi. Can Pseudonymity Really Guarantee Privacy? In S. Bellovin and G. Rose, editors, *9th USENIX Security Symposium, Denver, Colorado, USA, August 14-17, 2000*. USENIX Association, 2000.

[43] P. Rosso, F. Rangel, M. Potthast, E. Stamatatos, M. Tschuggnall, and B. Stein. Overview of PAN 2016—New Challenges for Authorship Analysis: Cross-genre Profiling, Clustering, Diarization, and Obfuscation. In *Experimental IR Meets Multilinguality, Multimodality, and Interaction. 7th International Conference of the CLEF Initiative (CLEF 2016)*, Berlin Heidelberg New York, Sept. 2016.

[44] C. Sanderson and S. Guenter. Short text authorship attribution via sequence kernels, markov chains and author unmasking: An investigation. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 482–491, July 2006.

[45] D. Sculley and C. E. Brodley. Compression and machine learning: A new perspective on feature space vectors. In *2006 Data Compression Conference (DCC 2006), 28–30 March 2006, Snowbird, UT, USA*, pages 332–332. IEEE Computer Society, 2006.

[46] E. Stamatatos. A Survey of Modern Authorship Attribution Methods. *Journal of the American Society for Information Science and Technology*, 60(3):538–556, Mar. 2009.

[47] E. Stamatatos, W. Daelemans, B. Verhoeven, M. Potthast, B. Stein, P. Juola, M. Sanchez-Perez, and A. Barrón-Cedeño. Overview of the Author Identification Task at PAN 2014. In *Working Notes Papers of the CLEF 2014 Evaluation Labs*, CEUR Workshop Proceedings. CLEF and CEUR-WS.org, Sept. 2014.

[48] E. Stamatatos, W. Daelemans, B. Verhoeven, M. Potthast, B. Stein, P. Juola, M. Sanchez-Perez, and A. Barrón-Cedeño. Overview of the Author Identification Task at PAN 2014. In *CLEF 2014 Evaluation Labs and Workshop – Working Notes Papers*.

[49] E. Stamatatos, W. D. amd Ben Verhoeven, P. Juola, A. López-López, M. Potthast, and B. Stein. Overview of the Author Identification Task at PAN 2015. In *CLEF 2015 Evaluation Labs and Workshop – Working Notes Papers*.

[50] E. Stamatatos, W. Daelemans, B. Verhoeven, P. Juola, A. López López, M. Potthast, and B. Stein. Overview of the Author Identification Task at PAN 2015. In *Working Notes Papers of the CLEF 2015 Evaluation Labs*, CEUR Workshop Proceedings.

[51] B. Stein, N. Lipka, and S. Meyer zu Eißen. Meta Analysis within Authorship Verification. In A. Tjoa and R. Wagner, editors, *5th International Workshop on Text-Based Information Retrieval (TIR 2008) at DEXA*, pages 34–39. IEEE, Sept. 2008.

[52] B. Stein, M. Potthast, and M. Trenkmann. Retrieving Customary Web Language to Assist Writers. In C. Gurrin, Y. He, G. Kazai, U. Kruschwitz, S. Little, T. Roelleke, S. M. Rüger, and K. van Rijsbergen, editors, *Advances in Information Retrieval. 32nd European Conference on Information Retrieval (ECIR 2010)*, volume 5993 of *Lecture Notes in Computer Science*, pages 631–635, Berlin Heidelberg New York, Mar. 2010.

[53] B. Stein, M. Hagen, and C. Bräutigam. Generating Acrostics via Paraphrasing and Heuristic Search. In J. Tsujii and J. Hajic, editors, *25th International Conference on Computational Linguistics (COLING 2014)*, pages 2018–2029. Association for Computational Linguistics, Aug. 2014.

[54] W. J. Teahan and D. J. Harper. Using compression-based language models for text categorization. In *Language modeling for information retrieval*, pages 141–165.

[55] C. Wu, X. Ren, F. Luo, and X. Sun. A Hierarchical Reinforced Sequence Operation Method for Unsupervised Text Style Transfer. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 4873–4883, July 2019.

[56] W. Xu, A. Ritter, B. Dolan, R. Grishman, and C. Cherry. Paraphrasing for style. In *Proceedings of COLING 2012*, pages 2899–2914, Mumbai, India, December 2012.

[57] Y. Zhao and J. Zobel. Searching with style: Authorship attribution in classic literature. In *Computer Science 2007. Proceedings of the Thirtieth Australasian Computer Science Conference (ACSC2007). Ballarat, Victoria, Australia, January 30 - February 2, 2007. Proceedings*, pages 59–68, 2007.

[58] Y. Zhao, J. Zobel, and P. Vines. Using Relative Entropy for Authorship Attribution. In H. T. Ng,

M. Leong, M. Kan, and D. Ji, editors, *Information Retrieval Technology, Third Asia Information Retrieval Symposium, AIRS 2006, Singapore, October 16-18, 2006, Proceedings*, volume 4182 of *Lecture Notes in Computer Science*, pages 92–105.

[59] R. Zheng, J. Li, H. Chen, and Z. Huang. A framework for authorship identification of online messages: Writing-style features and classification techniques. *Journal of the American Society for Information Science and Technology*, 57(3): 378–393, 2006.