Generalizing Unmasking for Short Texts

Janek Bevendorff[*] **Benno Stein**[*] **Matthias Hagen**[†] **Martin Potthast**[‡]

[*]Bauhaus-Universität Weimar
[†]Martin-Luther-Universität Halle-Wittenberg
[‡]Leipzig University

`<first>.<last>@uni-{weimar, leipzig}.de`
`<first>.<last>@informatik.uni-halle.de`

Abstract

Authorship verification is the problem of inferring whether two texts were written by the same author. For this task, unmasking is one of the most robust approaches as of today with the major shortcoming of only being applicable to book-length texts. In this paper, we present a generalized unmasking approach which allows for authorship verification of texts as short as four printed pages with very high precision at an adjustable recall tradeoff. Our generalized approach therefore reduces the required material by orders of magnitude, making unmasking applicable to authorship cases of more practical proportions. The new approach is on par with other state-of-the-art techniques that are optimized for texts of this length: it achieves accuracies of 75–80 %, while also allowing for easy adjustment to forensic scenarios that require higher levels of confidence in the classification.

1 Introduction

With advances in computational stylometry, determining the original authorship of unknown literary publications can be accomplished with near certainty by state-of-the-art authorship verification. If the source material is abundant and sufficiently many known publications exist, linking them together is hardly a challenge. But the playing field changes entirely if only fragments are available for verification, either because few known works of an author exist or because the text to be verified is only a few pages long. With classification results significantly above chance, yet far below certainty, verification approaches struggle to produce reliable results in short-text scenarios and thus lack real-world practicality for material far below book length. Even the unmasking approach by Koppel and Schler (2004), which otherwise proved to be one of the most ingenious and robust verification approaches, fails in this setting and can only deliver below-average performance compared to more specialized verification systems, which still display high uncertainty themselves with an error of up to 25 % (Stamatatos et al., 2015). At PAN 2015, individual texts of the English-language dataset had an average size of about 1.5 kB (less than 400 words), so it does not come as a surprise that none of the participants employed unmasking.

To tackle the uncertainty problem of authorship verification on short texts, we propose a generalized unmasking approach which prioritizes precision so as to verify authorship with reliable results while rejecting cases of low certainty. We also present a new open-source general-purpose unmasking framework as a highly-customizable implementation of our approach.[1]

2 Related Work

Authorship analysis has been practiced since the late 19th century (Bourne, 1897). Although mostly the narrower task of authorship attribution has been considered, where texts are attributed to a set of given authors, recently, authorship verification has been proposed as a more fundamental task. For long texts, unmasking by Koppel and Schler (2004) has since been established as a gold standard. Sanderson and Guenter (2006) showed that its performance is far worse for short texts, though, whereas their own model produced acceptable results with a minimum of 5,000 words per training text. Newer research has emerged through a series of shared tasks at PAN (Juola and Stamatatos, 2013; Stamatatos et al., 2014, 2015), which focused on shorter texts and achieved higher scores than unmasking. In general, however, the verification problem could not be solved to a point where results are highly reliable. The winner of the shared task at PAN 2015 (Bagnall, 2015) achieved an accuracy of 76 %, thus delivering a false decision in one in four cases while being unclear about which 24 %

[1] Code and data: https://github.com/webis-de/NAACL-19.

Proceedings of NAACL-HLT 2019, pages 654–659
Minneapolis, Minnesota, June 2 - June 7, 2019.

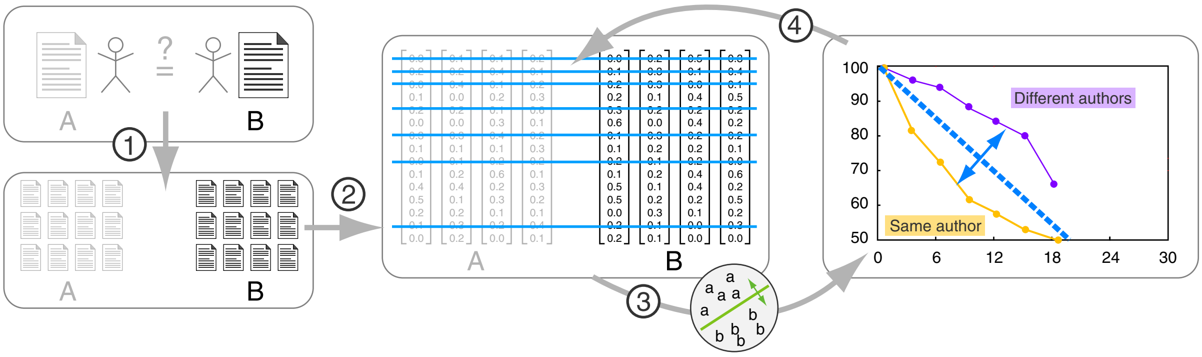

Figure 1: Schematic of the unmasking algorithm. Steps 1-4 are described in the text below. Dependent on whether the authors of texts A and B are the same or different, accuracy curves as exemplified can be expected.

are most likely to be incorrect. If applied in a real-world forensic scenario, such a verifier might give only hints as to whether two texts share authorship. Verification approaches based on text compression (Teahan and Harper, 2003; Khmelev and Teahan, 2003; Halvani et al., 2017) have been proposed, whose latest incarnations proved at least competitive to the approaches developed during the shared tasks; the state of the art in short-text authorship verification achieves around 75–80 % accuracy.

3 Unmasking for Short Texts

After reviewing the original unmasking algorithm, we introduce our generalization for short texts.

3.1 The Original Unmasking Algorithm

Unmasking as per Koppel and Schler (2004) 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 (see 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 %.

3.2 Generalization for Short Texts

While impressive as such, 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, 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. This is to 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:

1. From either text, create 30 chunks counting 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.

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 classifier is then used to classify curves generated in the same fashion from text pairs in the test set.

4 Evaluation

The data we use for our experiments is a collection of 180 text pairs (consisting of 90 *same-author* and 90 *different-authors* cases) for training and a similar set of 80 text pairs (consisting of 40 *same-author* and 40 *different-authors* cases) for testing. The texts were obtained from Project Gutenberg, comprise about 4,000 words each (23,000 characters), and were written by a total of 390 unique English-language authors. We took special care to select texts of similar genre and publication period to avoid accidental topic classification.

Since short-text authorship verification presents itself as such a difficult problem, it becomes all the more important to find a good measure for assessing the quality of a verifier. Due to the high uncertainty of many results, a standard two-class accuracy measure is not an optimal choice. Instead of trying to develop perfect verifiers, we can already build more useful tools today by optimizing for precision and sacrificing recall. Unfortunately, this approach—although more useful in general—does not perform well in a setting where accuracy is measured. In order to provide a more suitable evaluation quantity, PAN adopted the c@1 measure by Peñas and Rodrigo (2011):

$$\frac{1}{n} \cdot \left(n_{ac} + \frac{n_{ac}}{n} \cdot n_u\right) ,$$

where n denotes the number of problems, n_{ac} the number of correct answers and n_u the number of non-answers. The c@1 measure solves the uncertainty problem by rewarding non-answers in that it assigns them the same accuracy as the rest of the problems. All-correct answers still yield a score of 1, all-wrong or completely unanswered problem sets a score of 0. Hence, with this measure, a verifier is at liberty not to answer a given problem in case of doubt and still receive a reasonably good score, even if its overall sensitivity (or recall) is low. Most PAN participants did not exploit the c@1 measure to a larger extent, so c@1 scores of their submitted approaches are roughly the same as their accuracy (within margin of a few percent).

A problem with c@1 is that it is still designed for binary classification with equal weights for both classes. If we are primarily interested in whether two texts were written by the same author, but do not need a reliable decision in the other case, c@1 does not serve us well. For that reason, we propose as an alternative the $\mathrm{F}_{0.5}$ measure where we treat non-answers as false negatives:

$$\frac{(1 + 0.5^2) \cdot n_{tp}}{(1 + 0.5^2) \cdot n_{tp} + 0.5^2 \cdot (n_{fn} + n_u) + n_{fp}} ,$$

with n_{tp} denoting the number of true positives, n_{fn} the number of false negatives, and n_{fp} the number of false positives. As before, n_u is the number of unanswered problems. The parameter $\beta = 0.5$ of the F measure was chosen so as to weigh precision substantially higher than recall without diminishing its contribution entirely. Other values can be used depending on individual use cases. We call this specialized $\mathrm{F}_{0.5}$ measure $\mathrm{F}_{0.5\mathrm{u}}$.

4.1 Unmasking's Hyperparameters

The hyperparameters of unmasking have a direct and intuitive effect on its output. The most important hyperparameters are the number of chunks, the number of words per chunk, the size of the feature vectors, the number of feature removals per round, and the total number of averaged unmasking runs.

The degradation speed or curve slope is determined primarily by the number of chunks generated (more chunks result in generally shallower curves) and the amount of information lost in each round (shorter feature vectors or more removals cause the curves to plummet). Both parameters also control

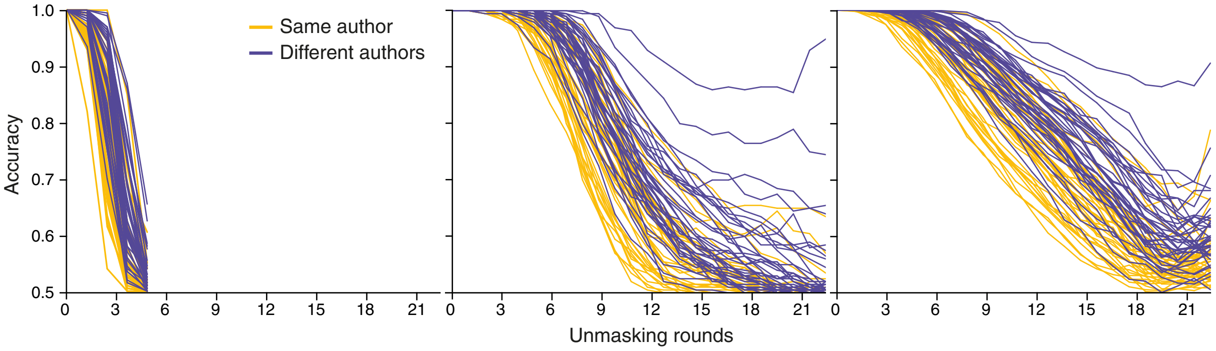

Figure 2: Unmasking curves on the test corpus with three different sets of hyperparameters. Each curve represents one text pair. *Left:* 100 features, 20 eliminations per round, 10 chunks of 1,000 words each. *Center:* 250 features, 10 eliminations, 10 chunks of 700 words each. *Right:* 250 features, 10 eliminations, 30 chunks of 700 words each. All curves are averages of 10 runs. Few features and chunks and many removals in a single round lead to curves dropping too quickly in straight lines. The center and right-hand configurations perform much better.

the granularity of the curves. Longer feature vectors and more chunks result in smoother curves, whereas very short vectors and very few chunks create larger straight-line segments. It is vital to find a good middle ground for both these parameters or else the curves will either altogether drop too quickly with all significant features eliminated in a single round, or runtime is wasted without benefit. Particularly removals and vector size need to be balanced so as to optimally capture enough of the highly significant features and not only the tail of the distribution. Furthermore, curves with insufficient granularity contain a lot less information and are often harder to distinguish, regardless of their slope. The number of words per chunk, on the other hand, appears to be mostly responsible for the slope of the first few initial rounds with larger chunks delaying the first major drop. Overall, this parameter's influence seems rather low, probably because we are only repeatedly sampling from an already small sample of the same distribution. Finally, the number of unmasking runs that are averaged smooths noisy curves and compensates for outliers. Here, more is generally better, but with diminishing returns beyond a certain point.

The hyperparameters given in Section 3 result from cross-validating several hundred configurations to find the most useful value ranges on our training data. Optimal choices can vary slightly between datasets, which leaves room for improvement. Figure 2 shows the differences between a select number of hyperparameter choices. We found that between 25 and 50 chunks, vector sizes of 250 to 400 features, and not fewer than 5, yet no more than 20 removals per round are necessary to achieve sufficient curve granularity and satisfying gradients. For chunk sizes, a large range between 300 and 1,000 words seems possible, though we found sizes of 500–700 words to work best without adding too much computational overhead. For stable classification, about 10 total runs need to be averaged, while 15–20 are still a sensible choice.

4.2 Verification Results

The overall performance of our approach is on par with other state-of-the-art verifiers, but we can improve our precision significantly by increasing the minimal distance a pair must have from the SVM hyperplane to be classified at all. We use this distance threshold as our confidence parameter c. Classification results at different values for c are shown in Table 1. At a threshold of 0.8, we can derive an answer for only 13.8 % of the cases, but are able to do so with a precision of 1.0, meaning all *same-author* classifications are correct. We can reduce the threshold down to a value of 0.4, at which we can classify about half the cases with a still all-correct answer set—but with lower confidence in the correctness. The actual numbers vary within a certain margin at low thresholds due to our random chunk generation. Thresholds below 0.4 will produce an increasing amount of false positives. The choice of c depends on the assurance level desired by the user. In medium- to high-assurance scenarios (where false positives are to be avoided, but not entirely critical), we recommend a threshold of 0.6 or higher. If false positives have to be entirely avoided, we recommend at least 0.7 or higher. On our dataset, results at $c \geq 0.8$ can be considered correct with near certainty. Cases which are not classifiable at the desired confidence threshold must be left undecided as "unknown authorship."

Hyperplane threshold c	**Classified cases [%]**	**Effectiveness**			
		Precision	Recall	$\mathrm{F}_{0.5\mathrm{u}}$	c@1
0.8	13.8	1.00	0.24	0.42	0.26
0.7	15.0	1.00	0.24	0.42	0.25
0.6	23.3	1.00	0.27	0.47	0.41
0.5	38.8	1.00	0.32	0.55	0.54
0.4	50.0	1.00	0.39	0.64	0.66
0.1	91.2	0.82	0.54	0.74	0.76
0.0	100.0	0.83	0.59	0.76	0.73

Table 1: Unmasking performance on our test data at various confidence thresholds. Recall was calculated after assigning all non-decisions the negative class. $\mathrm{F}_{0.5\mathrm{u}}$ and c@1 diverge significantly at high thresholds with an increasing class balance skew.

Approach	**Precision**	**Recall**	$\mathbf{F}_{0.5\mathbf{u}}$	**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 2: Performance comparison with the state of the art in short-text authorship verification ($c = 0.1$ in generalized unmasking). Differences between the first three are non-significant.

We compare the performance of our generalized unmasking approach to the two state-of-the-art short-text verification approaches by Bagnall (2015) and Halvani et al. (2017). Since no ready-to-use implementation of the latter approach is available, we reimplemented it using the PPMd compression algorithm and trained a random forest with 100 trees on the two suggested similarity measures Compression-based Cosine (CBC) and the Chen-Li metric (CLM). For a fairer comparison, we use a confidence threshold $c = 0.1$ for unmasking as this results in about the same amount of c@1 "optimization" as employed by Bagnall, although our $\mathrm{F}_{0.5\mathrm{u}}$ is higher at $c = 0$. The results of our performance comparison are shown in Table 2. The precision of our new generalized unmasking is the highest with slightly worse c@1 and $\mathrm{F}_{0.5\mathrm{u}}$ compared to Bagnall and CLM due to a lower recall. CBC appears to score the worst, although Halvani et al. found it to be best-performing. Overall, however, all approaches perform equally well and (treating non-answers as incorrect) no actual difference between their (binary) decision quality can be inferred from a McNemar test between unmasking and Bagnall ($\chi^2 = 0.35$, $p = 0.56$) or unmasking and Halvani et al. with CLM ($\chi^2 = 0.83$, $p = 0.36$).

In terms of runtime, compression performs best with under a minute on the whole corpus due to the optimized C implementation of the compressor, followed by our new generalized unmasking implementation with about 2–3 minutes. Bagnall runs the longest with 4:45 hours on all 80 cases in the corpus. It is worth noting that, unlike unmasking and compression models, Bagnall's approach is not intrinsic and needs other texts from the input corpus to arrive at a decision for the single pair in question, which is a valid approach, but can effectively exploit potential biases in the corpus. We further noticed that compression appears to balance precision and recall more than the other approaches, whereas generalized unmasking heavily prioritizes precision, which is generally more preferable in a real-world scenario: here, false positives can have dire consequences, such as a wrong conviction. Another advantage of generalized unmasking is that it allows for easy optimization of all hyperparameters compared to the other approaches which are more of a blackbox. By building ensembles with other machine learning models, different features, chunk sizes, etc., we expect further improvements, but leave this for future work.

4.3 Reproducibility

To enable our research on short-text unmasking, we developed an extensive general-purpose unmasking framework for running any kind of unmasking experiment with a plethora of different features, parameters, and aggregations for final or partial results from different runs. Each unmasking run comes with a detailed job configuration allowing for easy reproduction of previous experiments. We published the source code of our framework and all data used in our research under an open-source license alongside this paper.

5 Conclusion

We have successfully generalized and applied unmasking to short texts establishing another state-of-the-art verification approach for scenarios in which only little source material is available. This new unmasking approach prioritizes precision to deliver highly-reliable decisions even though short texts naturally show a high amount of uncertainty as a result of their low stylistic information content. We compared our approach to other state-of-the-art authorship verification systems specialized for short texts and can produce competitive results with low computational effort and high prospect of further optimization in the future.

References

Douglas Bagnall. 2015. Author identification using multi-headed recurrent neural networks—Notebook for PAN at CLEF 2015. In *CLEF 2015 Evaluation Labs and Workshop – Working Notes Papers*.

Edward Gaylord Bourne. 1897. The authorship of the federalist. *The American Historical Review*, 2(3):443–460.

Oren Halvani, Christian Winter, and Lukas Graner. 2017. Authorship verification based on compression-models. ArXiv 1706.00516.

Patrick Juola and Efstathios Stamatatos. 2013. Overview of the author identification task at PAN 2013. In *CLEF 2013 Evaluation Labs and Workshop – Working Notes Papers*.

Dmitry V. Khmelev and William John Teahan. 2003. A repetition based measure for verification of text collections and for text categorization. In *Proceedings of SIGIR 2003*, pages 104–110.

Moshe Koppel and Jonathan Schler. 2004. Authorship verification as a one-class classification problem. In *Proceedings of ICML 2004*, pages 1–7.

Anselmo Peñas and Álvaro Rodrigo. 2011. A simple measure to assess non-response. In *Proceedings of ACL 2011*, pages 1415–1424.

Conrad Sanderson and Simon Guenter. 2006. Short text authorship attribution via sequence kernels, Markov chains and author unmasking: An investigation. In *Proceedings of EMNLP 2006*, pages 482–491.

Efstathios Stamatatos, Walter Daelemans, Ben Verhoeven, Patrick Juola, Aurelio López-López, Martin Potthast, and Benno Stein. 2015. Overview of the author identification task at PAN 2015. In *CLEF 2015 Evaluation Labs and Workshop – Working Notes Papers*.

Efstathios Stamatatos, Walter Daelemans, Ben Verhoeven, Martin Potthast, Benno Stein, Patrick Juola, Miguel A. Sanchez-Perez, and Alberto Barrón-Cedeño. 2014. Overview of the author identification task at PAN 2014. In *CLEF 2014 Evaluation Labs and Workshop – Working Notes Papers*.

William J. Teahan and David J. Harper. 2003. Using compression-based language models for text categorization. In *Language Modeling for Information Retrieval*, pages 141–165.