Bootstrap Distance Imposters: High precision authorship verification with improved interpretability*

Ben Nagy[1]

[1]Institute of Polish Language, Polish Academy of Sciences (IJP PAN)
al. Mickiewicza 31
Kraków, Poland

Abstract

This paper describes an update to the open-source Python implementation of the General Imposters method of authorship verification by Mike Kestemont et al. The new algorithm, called Bootstrap Distance Imposters (henceforth BDI), incorporates a key improvement introduced by Potha and Stamatatos, as well as introducing a novel method of bootstrapping that has several attractive properties when compared to the reference algorithm. Initially, we supply an updated version of the Kestemont et al. code (for Python 3.x) which incorporates the same basic improvements. Next, the two approaches are benchmarked using the problems from the multi-lingual PAN 2014 author identification task, as well as the more recent PAN 2021 task. Additionally, the interpretability advantages of BDI are showcased via real-world verification studies. When operating as a summary verifier, BDI tends to be more conservative in its positive attributions, particularly when applied to difficult problem sets like the PAN2014 *en_novels*. In terms of raw performance, the BDI verifier outperforms all PAN2014 entrants and appears slightly stronger than the improved Kestemont GI according to the PAN metrics for both the 2014 and 2021 problems, while also offering superior interpretability.

Keywords

authorship verification, stylometry, bootstrapping

1. Introduction

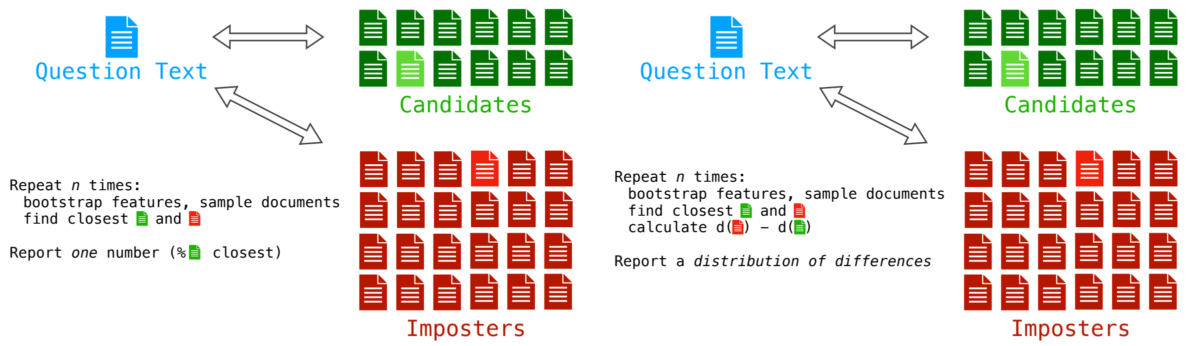
A common task in the field of stylometry is to attempt to resolve questions of authorship. In "authorship verification" the basic question is "how likely is it that a given document is by a specified author?". Since the answers to these questions can have important repercussions, whether legal, literary, or social, it is important that the methods employed by practitioners be reliable; but in order for the results to be believed they should also be transparent and interpretable. The General Imposters method (henceforth GI), originally formulated by Koppel and Winter in [9], has become one of the standard methods in stylometry for authorship verification. After strong performances in the PAN 2013 and 2014 competitions, it was implemented and used by M. Kestemont et al. in an influential study [6], improved by Potha and Stamatatos in 2017 [14], and is now available in the well-known R package *stylo* [4].

In this paper I describe an update to the open-source Python implementation by Kestemont et al. [5] called Bootstrap Distance Imposters (henceforth BDI) which incorporates several of the improvements proposed since the last release, as well as introducing a novel method of bootstrapping that has several attractive properties when compared to the reference algorithm. The two approaches are benchmarked using the problems from the PAN 2014 author identification task [16], and some additional properties are showcased via real-world case studies.

CHR 2024: Computational Humanities Research Conference, December 4–6, 2024. Aarhus, Denmark

*I am grateful to various members of the Computational Stylistics Group https://computationalstylistics.github.io/ for their ideas and support. Particular thanks are due to Mike Kestemont for a kind but careful review of the initial draft; subsequent errors are my own. I thank the anonymous CHR reviewers for their thoughtful comments, some of which could not be implemented in time for proceedings but offer exciting potential for further work.

benjamin.nagy@ijp.pan.pl (B. Nagy)
https://github.com/bnagy/bdi-paper (B. Nagy)
0000-0002-5214-7595 (B. Nagy)



Standard General Imposters | Bootstrap Distance Imposters

Figure 1: A simplified comparison of the operation of 'standard' General Imposters and Bootstrap Distance Imposters.

2. Motivation and Design

The Imposters method of authorship verification examines a *question text* in order to determine whether the text was written by a *candidate author*. To do this, it performs bootstrap comparisons of the question text to candidate texts and several (perhaps very many) *imposter texts* which should ideally be chosen to be similar in genre, topic and register. If the question text is markedly more similar to the candidate author's 'style' than to any of the imposters then we infer, with some level of statistical likelihood, that it was authored by the candidate. The method of Imposters can be used with any features that reflect style, although it is most commonly applied to distances between character *n*-gram or word frequency vectors.

Note that while I speak of 'statistical likelihood' I am carefully avoiding the words 'probability' and 'confidence'. The GI method is, in machine learning terms, an ensemble classifier. These classifiers regularise well, but while they produce a real-valued output, it is problematic to interpret this number as a probability. Many factors can make the verification results less reliable—the lengths of the texts, the language in which they are written, the closeness of the imposter texts (dissimilar texts make a positive attribution less convincing), and the amount of available data (a lack of comparison data risks bias). Verification problems in the real world are seldom under ideal conditions, and there is no magical formula by which the uncertainties imposed by the problem setting can be convincingly rendered as a forensic probability. Never the less, the Imposters method has a well deserved reputation for robust, understandable results, even in the face of severe limitations (for example in the length of samples or the availability of suitable imposters).

Figure 1 attempts an intuitive explanation of the basic operation of standard GI and the key modification used in BDI. The output of the Kestemont GI classifier is a percentage of binarized 'votes' (the number of times a candidate text was closer than an imposter). In contrast, the raw output from the BDI algorithm is a bootstrapped distribution of differences. At each step, the distance (with a bootstrapped feature set) between the candidates and the imposters is recorded, using any vector distance measure $d : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}$. If the candidates are further, the difference between the distances is negative, if closer it is positive. If these individual distances follow a Gaussian distribution (which is a reasonable prior expectation) then their difference is also Gaussian. Expressing the results this way has some advantages. The first is that we can differentiate a negative result (not the candidate) as either 'none of the above' or 'more like an imposter' (the true author is in the imposters set). A 'none of the above' result would have a statistically expected distance of zero (equally unlike the candidate and the imposters), and so we would see a distribution centred around 0.[1] On the other hand, 'more like an imposter' results show distributions centred around a negative value (examples of this can be seen in Section 4 below). The

[1]Note carefully that this is a one-way implication—a true author that is neither the candidate nor one of the imposters should have a distance distribution centred around zero, but not all such distributions guarantee that the true author is not among the imposters.

other advantage is that for strong positives, we have additional data about the match. Distributions centred around larger positive numbers are better matches, but distributions with high variance show more feature dependence (since the strength of the match varies greatly depending on the bootstrap feature sets). In summary, positive matches (with most or all of the probability mass above zero) can be much more meaningfully compared.

It is worth noting here that the overall best performing method at PAN 2014 by Khonji and Iraqi [8] also modified the classic GI algorithm to utilize the distance between vectors (in that case the relative distance of the test vector to candidates vs imposters was considered as part of the decision function for a 'standard' voting-based classifier), so this paper is not the first to recognise the value of this additional information.

2.1. Binary Classification

Based on the BDI algorithm, which outputs a distribution, it is obviously useful to have a summary statistic that can be interpreted as evidence for authorship verification tasks. For this paper I used a simple approach that considers the amount of probability mass that lies above 0. If every test is closer to a candidate than an imposter then the result will be 1, if every test is more like an imposter, it will be 0, etc. This is implemented simply as the inverse percentile of (a distance of) 0. Thus armed with a method that outputs a 'probability-like' result in $[0, 1]$, I wrapped the code in a classifier that follows `scikit-learn` [13] conventions like `fit()` and `predict_proba()` and evaluated the BDI classifier directly against the updated Kestemont GI `Order2Verifier` using the PAN shared tasks from 2014 and 2021. This provided a convenient benchmark, and also an opportunity to compare the results against a number of other verification approaches.

2.2. Score Shifting

The C@1 metric introduced in PAN 2014 rewards (or at least penalises less harshly) classifiers that choose not to answer some problems. This leads naturally to algorithms that use the training data to define classifier output ranges that will be assigned to 0.5 (indicating an unanswered problem). In the case of classic GI, this means that classifier scores (vote percentages) within certain ranges will be rectified to precisely 0.5, while positive and negative classifications are shifted to the ranges above and below that value. This (hopefully) improves the C@1 score as compared to basic accuracy. In the PAN 2020–21 competitions, a similar measure was used called $F_{0.5u}$ (introduced in [2]).

The score shifting method implemented in Kestemont GI attempts to produce something more like a probability by linearly scaling the output values. The code defines an upper and lower bound for the unanswered region, `p1` and `p2` The scaling code (in Python) looks like this:

```python
for score in list(scores):
    if score <= p1:
        new_scores.append(rescale(score, min(scores), max(scores), 0.0, p1))
    elif score >= p2:
        new_scores.append(rescale(score, min(scores), max(scores), p2, 1.0))
    else:
        new_scores.append(0.5)
```

Scores below `p1` are scaled to $[0, \texttt{p1})$, scores above `p2` are scaled to $(\texttt{p2}, 1]$, and the rest are coerced to 0.5. There is an issue with this scaling algorithm, however. Since `p1` and `p2` are chosen by grid search to maximise the PAN score, the score shifter sometimes fits values for `p2` that are well below 0.5. This can lead to decisions that are defined as positive (since they are above `p2`) being scaled to below 0.5, where they are evaluated by the scoring metrics as a negative result (and scored as such). In the updated code, I modified the shifting code to scale more simply to $[0, 0.5)$ (negative), 0.5 (unanswered), and $(0.5, 1]$ (positive). This does not retain the global distributional properties of the original results (as implemented in [6]).

```
for score in scores:
    if score <= p1:
        new_scores.append( rescale(score, orig_min=0, orig_max=p1, new_min=0.0,
         ↪ new_max=0.499) )
    elif score >= p2:
        new_scores.append( rescale(score, orig_min=p2, orig_max=1,
         ↪ new_min=0.501, new_max=1.0) )
    else:
        new_scores.append(0.5)
```

Based on the evaluation problems, the BDI algorithm is not as sensitive to this score shifting, deriving only a modest benefit from fitting. The fitting process raises natural questions about the representativeness of the training data, and also causes some problems in domains that suffer from limited data availability (where it can be hard to sacrifice data for training). In these circumstances, BDI works well with manual score shifting, allowing the user to choose a confidence level empirically.

2.3. Changes to Kestemont GI

As is the nature of software, the code in the repository documenting the GI algorithm and for the related work on the Caesarian corpus no longer ran. I reworked the code slightly, and made the following small changes, which are available in my own repository [11].

- Update the code to work with Python 3 (these minimal changes have been incorporated into the original repository based on a PR);
- Implement a fast 'nini' metric (fuzzy Jaccard similarity) as described in [12];
- Implement the Potha & Stamatatos 'ranking' improvement for the consensus score[2] described in [14];
- Remove most non-core code;
- Modify the score-shifting algorithm, as described above.

3. Evaluation

The BDI classifier was compared head-to-head with the updated Kestemont `Order2Verifier` on the full PAN 2014 evaluation corpus, which is a challenging set of varied authorial styles across four languages.[3] Additionally, both verifiers were evaluated below using the PAN 2021 problems, since that competition included some deep-learning approaches, discussed further below. For this evaluation I used character n-grams, since that feature universe is a generally reliable and uncontroversial choice for modern stylometric work. There may be feature universes that perform better, or features that work better for a specific task, but character n-grams are 'solid if boring'. Likewise, the n-gram frequencies are z-scaled (based only on the training variances) since this is the standard approach. I tested two n-gram configurations, 2,3,4-grams and 2,3,4,5-grams, 10,000 max features, with fitted and manual score shifting. For the `Order2Verifier` bootstrapping was performed at 50% (5000 random features per bootstrap iteration) since that is the configuration used in [6]; for BDI, bootstrapping was performed at 33%. Finally, for the distance metric used to determine 'closeness' at each step I tested the cosine distance (the most traditional choice) and the minmax (Ružička) metric promoted by Kestemont et al.

[2]Instead of a strict 1 (candidate closest) or 0 (candidate not closest), Potha & Stamatatos proposed a score improvement based on the ordinal rank of the closest candidate, so if a candidate was the second-closest, the score for that iteration would be $\frac{1}{2}$. The same paper proposed a distance-based culling method to select more relevant imposters, but this was not implemented because of poor Big-O complexity.

[3]There is a small inconsistency that I was unable to resolve—the only copy of the verification problems I could find were archived in the Kestemont repository, but they are apparently missing 50 of the 'Dutch Reviews' evaluation problems, so there are a total of 746, versus 796 reported in the PAN 2014 wrapup report.

Table 1
Global micro-average results. PAN21 and PAN14 refer to the overall evaluation metrics used in each competition. PAN14-U(nranked) is the overall PAN 2014 score for each classifier when run without the ranking improvement.

Verifier	Vectorizer	Shifter	AUC	C@1	$F_{0.5u}$	F_1	Brier	Prec	PAN21	PAN14	PAN14-U
BDI, Cosine	2,3,4,5-grams	fitted	0.718	0.696	0.686	0.616	0.731	0.849	0.689	0.500	0.465
		manual	0.719	0.654	0.541	0.482	0.717	0.944	0.623	0.470	0.426
	2,3,4-grams	fitted	0.729	0.691	0.678	0.611	0.735	0.821	0.689	0.504	0.463
		manual	0.727	0.655	0.525	0.478	0.724	0.952	0.622	0.476	0.424
BDI, Minmax	2,3,4,5-grams	fitted	0.731	0.703	0.694	0.625	0.738	0.852	0.698	0.514	0.460
		manual	0.732	0.665	0.555	0.493	0.722	0.936	0.633	0.487	0.414
	2,3,4-grams	fitted	**0.737**	**0.706**	**0.707**	0.632	0.739	0.846	**0.704**	**0.520**	0.458
		manual	0.731	0.663	0.551	0.500	0.728	**0.956**	0.634	0.485	0.412
Kestemont GI, Cosine	2,3,4,5-grams	fitted	0.723	0.673	0.660	0.650	0.784	0.788	0.698	0.487	0.457
		manual	0.689	0.548	0.500	0.699	0.783	0.918	0.644	0.378	0.428
	2,3,4-grams	fitted	0.715	0.664	0.645	0.642	0.780	0.778	0.689	0.475	0.456
		manual	0.693	0.548	0.498	0.714	0.784	0.926	0.647	0.380	0.414
Kestemont GI, Minmax	2,3,4,5-grams	fitted	0.723	0.674	0.662	0.634	0.782	0.835	0.695	0.487	0.470
		manual	0.700	0.568	0.513	0.700	0.785	0.912	0.653	0.398	0.444
	2,3,4-grams	fitted	0.721	0.666	0.652	0.649	0.784	0.809	0.694	0.480	**0.479**
		manual	0.703	0.562	0.515	**0.719**	**0.788**	0.929	0.658	0.395	0.435

Table 2
BDI 2,3,4,5-grams, Minmax, Manual Shifter, U_{count} is the number of unanswered problems. Summary scores between U_{low} and U_{high} are left unanswered (changed to 0.5).

Corpus	Tests	U_{count}	U_{low}	U_{hi}	Prec	AUC	C@1	$F_{0.5u}$	F_1	Brier	PAN21	PAN14
du_essays	96	20	0.110	0.890	0.951	0.952	0.919	0.871	0.963	0.916	0.924	0.875
du_reviews	50	10	0.110	0.890	0.667	0.594	0.552	0.250	0.190	0.630	0.443	0.328
en_essays	200	27	0.110	0.890	1.000	0.638	0.568	0.231	0.141	0.634	0.442	0.362
en_novels	200	28	0.110	0.890	1.000	0.667	0.587	0.236	0.148	0.632	0.454	0.392
gr_articles	100	34	0.110	0.890	0.818	0.856	0.697	0.455	0.562	0.800	0.674	0.597
sp_articles	100	35	0.110	0.890	0.963	0.882	0.810	0.751	0.912	0.856	0.842	0.714

Table 3
Kestemont 2,3,4,5-grams, Minmax, Fitted Shifter

Corpus	Tests	U_{count}	U_{low}	U_{hi}	Prec	AUC	C@1	$F_{0.5u}$	F_1	Brier	PAN21	PAN14
du_essays	96	17	0.277	0.777	0.935	0.953	0.932	0.881	0.966	0.918	0.930	0.888
du_reviews	50	11	0.125	0.285	0.667	0.624	0.561	0.506	0.500	0.746	0.587	0.350
en_essays	200	19	0.527	0.679	0.556	0.559	0.498	0.273	0.182	0.665	0.435	0.278
en_novels	200	42	0.089	0.241	0.864	0.683	0.641	0.629	0.594	0.774	0.664	0.438
gr_articles	100	29	0.473	0.849	0.857	0.839	0.735	0.634	0.720	0.820	0.750	0.617
sp_articles	100	13	0.428	0.643	0.849	0.934	0.848	0.821	0.882	0.881	0.873	0.792

in their original paper. Consistent with those results, the minmax metric appears generally superior (see Table 1). The fully reproducible testing code is available in the supplementary repository [10]. Several metrics are reported, some of which (C@1, $F_{0.5u}$) are rare outside the PAN competitions, along with the overall evaluation scores for PAN 2014 (AUC×C@1) and PAN 2021 (mean of the five balanced measures).

3.1. Results vs PAN 2014

The format of the PAN 2014 problems makes it straightforward to apply the BDI and GI verifiers and directly compare the results. As can be seen from Table 1, the performance of the BDI classifier is consistently strong, with only a small boost in PAN score resulting from fitting the score shifter. When evaluated as a global micro-average (as done in the competition), the 'best' BDI verifier outperformed

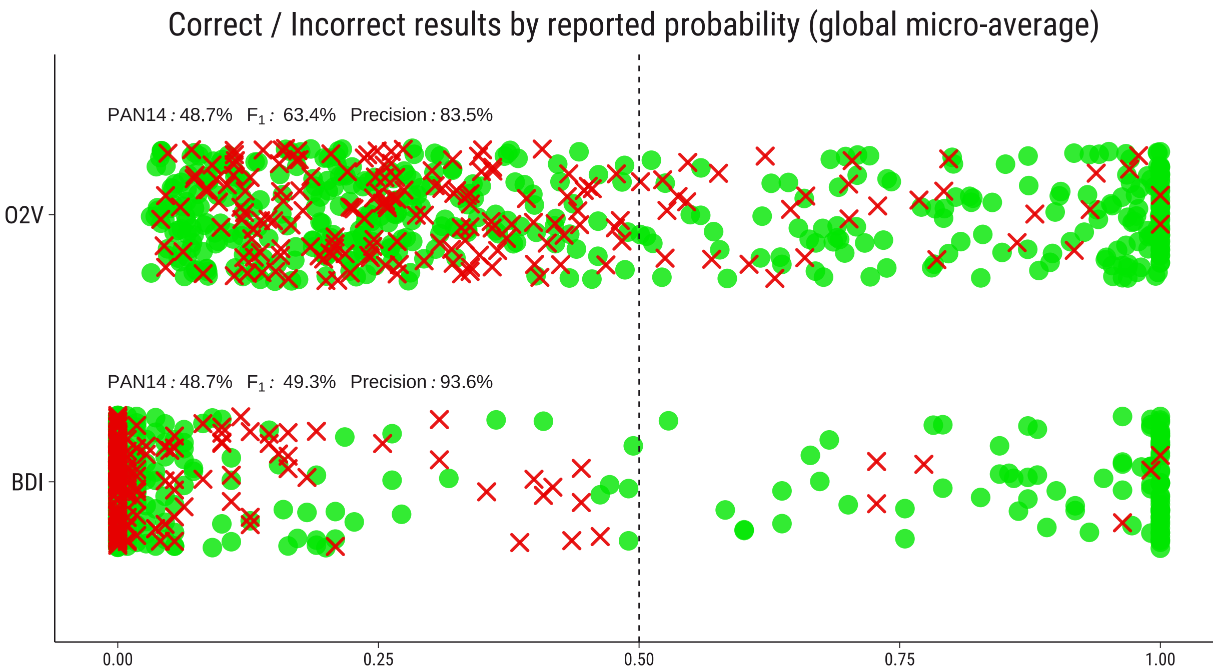


Figure 2: A comparison of the correct and incorrect results by reported probability for BDI (manual shifter, 2,3,4,5-grams, minmax) vs Order2Verifer (fitted shifter, 2,3,4,5-grams, minmax) on the full PAN 2014 set. Performance metrics are global micro-averages, $n = 746$.

both updated Kestemont GI and the winning PAN 2014 entrant. It must be noted, however, that the individual corpus results are not dominated by either verifier but vary considerably according to evaluation strategy and corpus. The strength of the BDI classifiers, however, is twofold: first, they do not really require any training corpus, and second, the BDI approach is high precision (it yields very few false positives), making it a conservative verifier whose positive results are reliable (at the cost of more false negatives). This can be seen clearly in Figure 2 in which the best performing GI classifier is compared to a manually fitted BDI classifier (results between 11% and 89% are left unanswered) using the same features and metrics. This is not the best performing BDI verifier: it was chosen because these verifiers have an almost identical overall PAN14 score but quite different detailed characteristics. The results for each subcorpus are broken down in more detail in Tables 2 & 3. In those tables, the respective results for the *en_novels* set are of particular interest. This sub-corpus was extremely challenging, causing problems for all entrants due to the shared-genre nature of Lovecraftian horror and the unifying force of its pervasive style (explained in more detail in [16, p. 882]). BDI underperformed here according to the PAN metrics, but did so conservatively, with a perfect precision score of 1.0.

3.2. Testing vs PAN 2021

Since the state of the art in authorship verification is now exploring the possibilities offered by deep learning, the BDI verifier (and the `Order2Verifier`) were also evaluated using the data from the PAN 2020–2021 shared task. The final evaluation report from that task is available in [7]. The 2020–21 tasks used English-language fan-fiction, which is available in vast amounts. Although the huge amount of training data and the limitation to English makes the problems somewhat less interesting, it is interesting to know how much we 'lose' by using simple machine learning approaches that can be widely applied as compared to cutting-edge techniques applied to best-case scenarios. The large amount of training data (the **large** training set contained 176,000 total text pairs, and even these were synthetically augmented by some teams!), as well as the more recent date, meant that the results here were dominated by complex deep learning models, with the winning entry being an extremely impressive siamese network using four subcomponents [3]. It is not possible to provide a strict 'apples to apples' comparison, since the entrants were required to assess each problem as an atomic pairwise determi-

Table 4
Fitting Results: PAN 2021 (small training corpus), 1000 problems

Verifier	Vectorizer	Shifter	U_{count}	U_{low}	U_{hi}	Prec	AUC	C@1	$F_{0.5u}$	F_1	Brier	PAN21
BDI, Cosine	2,3,4,5-grams	fitted	94	0.089	0.661	0.960	0.942	0.911	0.902	0.917	0.910	0.917
		manual	131	0.110	0.890	0.979	0.939	0.909	0.893	0.918	0.906	0.913
	2,3,4-grams	fitted	95	0.205	0.741	0.970	0.947	0.920	**0.909**	0.923	0.914	0.923
		manual	146	0.110	0.890	0.979	0.944	0.916	0.892	0.931	0.909	0.918
BDI, Minmax	2,3,4,5-grams	fitted	101	0.169	0.822	0.977	0.941	0.913	0.906	0.916	0.908	0.917
		manual	136	0.110	0.890	0.979	0.937	0.907	0.891	0.918	0.906	0.912
	2,3,4-grams	fitted	120	0.125	0.724	0.973	**0.951**	**0.924**	0.905	0.935	**0.918**	**0.927**
		manual	149	0.110	0.890	0.979	0.948	0.919	0.894	0.936	0.914	0.922
Kestemont GI, Cosine	2,3,4,5-grams	fitted	119	0.312	0.777	0.966	0.944	0.913	0.898	0.924	0.909	0.918
		manual	328	0.110	0.890	0.981	0.920	0.857	0.830	0.964	0.875	0.889
	2,3,4-grams	fitted	130	0.312	0.741	0.966	0.947	0.921	0.898	0.936	0.911	0.923
		manual	346	0.110	0.890	0.981	0.916	0.847	0.819	0.966	0.873	0.884
Kestemont GI, Minmax	2,3,4,5-grams	fitted	104	0.393	0.786	0.973	0.943	0.915	0.906	0.921	0.912	0.920
		manual	328	0.110	0.890	0.984	0.919	0.857	0.831	0.965	0.875	0.889
	2,3,4-grams	fitted	132	0.312	0.741	0.971	0.950	**0.924**	0.902	0.940	0.914	0.926
		manual	338	0.110	0.890	**0.984**	0.924	0.858	0.831	**0.972**	0.876	0.892

nation (were these two texts written by the same author). For the Imposters methods in general, at least one sample from the candidate author is required, and a pool of imposters is used at each step. However, the evaluation was done as fairly as possible. Although the BDI and GI verifiers are not 'trained' per-se, both the score shifting parameters for the fitted shifters as well as the variances used in z-scaling were derived from a subset of the **small** training corpus. This is in spirit with the real PAN competition where the final evaluation dataset was fully blinded, unlike in previous years. The amount of training data employed was very small, using just 4256 texts from the 'small' training set, which contained 106,000 texts. The final evaluation used texts from the 2021 evaluation set (19999 pairs) which contained no texts from authors in the training sets. To fairly assess AUC, a roughly even split of positive and negative determinations are required. In addition, since the 2021 shared task was focused on previously-unseen authors, unseen authors were included as noise in both the comparison set (candidate profiles) and the evaluation set (verification problems). The evaluation set that was eventually used contained 10,152 problems, and was broken down as follows:

- 5076 problems from authors with at least one other text, tested against the true label;
- 2538 problems from authors with at least one other text, tested against a randomly chosen false label;
- 2538 problems from unseen authors, tested against a random false label.

The candidate/comparison set is composed as follows:

- 1692 authors who are never seen again, as noise;
- 7451 authors with one comparison sample (who might appear against a true or false label);
- 905 authors with 5 comparison samples;
- 94 with 11;
- 7 with 19;
- 3 with 29.

As can be seen, the large majority of authors have only one comparison candidate. The unnatural distribution of counts for texts per author is an artifact of the PAN data, and is presumably related to the corpus compilation process. The evaluation process used all of the evaluation texts by authors with two or more total samples, but the bulk of the singleton texts were not used. The full evaluation process is available as commented Jupyter notebooks at the accompanying repository.

Overall, the best performers in training were fitted versions of the 2,3,4-gram Kestemont GI and BDI verifiers, once again with the minmax metric. The full results can be seen in Table 4. Since the

Table 5
Evaluation Results: PAN 2021, 10,152 test problems

Verifier	Shifter	U_{count}	U_{low}	U_{hi}	Prec	AUC	C@1	$F_{0.5u}$	F_1	Brier	PAN21
	fitted	1326	0.125	0.724	0.975	0.922	0.881	0.872	0.882	0.885	0.889*
BDI, Minmax	fitted (optimal)	1128	0.053	0.393	0.943	**0.924**	**0.889**	**0.876**	0.896	**0.892**	**0.896**
	manual	1815	0.110	0.890	0.984	0.918	0.870	0.843	0.877	0.878	0.877
	fitted	1385	0.312	0.741	0.974	0.923	0.874	0.865	0.874	0.888	0.885
Kestemont GI, Minmax	fitted (optimal)	1433	0.196	0.527	0.940	**0.924**	0.883	0.863	0.898	0.891	0.892
	manual	3578	0.110	0.890	**0.987**	0.892	0.820	0.774	**0.931**	0.860	0.855
BDI, Minmax , unranked	fitted	1469	0.125	0.724	0.982	0.915	0.859	0.847	0.848	0.868	0.867
Kestemont GI, Minmax , unranked	fitted	940	0.312	0.741	0.982	0.917	0.841	0.855	0.809	0.858	0.856

bootstrapping process was extremely time consuming on the full evaluation set, only the 2,3,4-gram vectorizer was used. The best performing BDI verifier posted a final overall score of 0.889, using the PAN 2021 metric (the mean of several accuracy measures, calculated using the official evaluation code). This would place it near the bottom of the four "strong runner ups" [7, p. 7], still comfortably outperforming the machine-learning benchmarks and 5 human teams. It seems reasonable to assume that a more sophisticated feature vector, such as the one used in weerasinghe21 [17], would improve the performance of both Order2Verifier and BDI. By comparison, the final score for boenninghoff21 was 0.9545. Once again, it is clear from Tables 4 and 5 that the fitting is nice to have in terms of the overall PAN21 score (which uses several balanced accuracy measures), but in fact reduces raw precision, which is relevant where research problems require the smallest chance of false positives.

As further exploration, some differential / ablation results are included in Table 5. The results under the shifter method 'fitted (optimal)' calculate the optimal post-hoc p1 and p2 (as discussed above) based on the results from the evaluation itself—in other words it shows the advantage if the training regime were to perfectly represent the probability distribution of the true problems. As can be seen, this is quite minor (an overall score of 0.896 vs 0.889), which is encouraging. However, the relatively strong performance of the manual shifter with the BDI verifier, as well as its higher precision, seems to suggest that the small increase in some of the measures does not warrant the methodological uncertainty (in terms of representativeness and bias) added by the fitting process in general. Additionally, it can be seen that the score shifting optimisation is very sensitive to the evaluation measure, incurring quite a large precision penalty—not a component of the target metric—in order to improve the overall score. Finally, the new addition of 'ranking' was evaluated by ablation. As in the PAN 2014 results (Table 1, PAN14-U), the ranking regularisation appears to offer a small but consistent benefit to AUC and F-measures, which is impressive considering how few of the authors (about 10%) had more than one comparison sample. The unranked verifiers, however, showed even higher precision which may need to be considered for some applications.

4. Showcase

This section refers to two verification studies in which BDI has been applied, one of which is, at the time of writing, still in press. These figures are not full summaries of the research, but simply illustrate some of the features of the BDI method that I believe to be useful. As mentioned above, the output of the BDI algorithm is a distribution of distances, not a summary statistic. These examples attempt to show that examining the full distribution conveys extra information and can improve our intuition and confidence in the analysis.

Figures 3 & 4 are from an analysis of several poems attributed to Ovid. The aim here was to provide evidence for the genuineness of the *Nux*, but of more interest in this context is the analysis of the *Consolatio ad Liviam*. The *Consolatio* was once considered to be a genuine work of Ovid, but is now accepted by most scholars to be a first-century imitation. By using BDI we attempted to show that

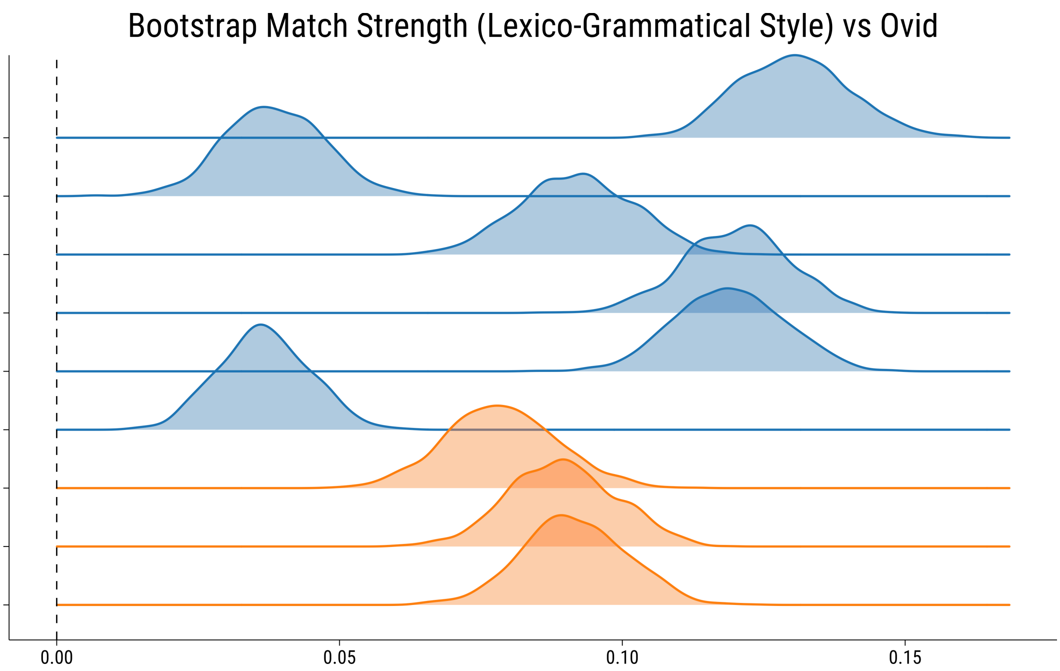


Figure 3: A BDI comparison of several works attributed to Ovid using lexico-grammatical features (character *n*-grams). The *Consolatio Ad Liviam* is now considered to be a first-century imitation.

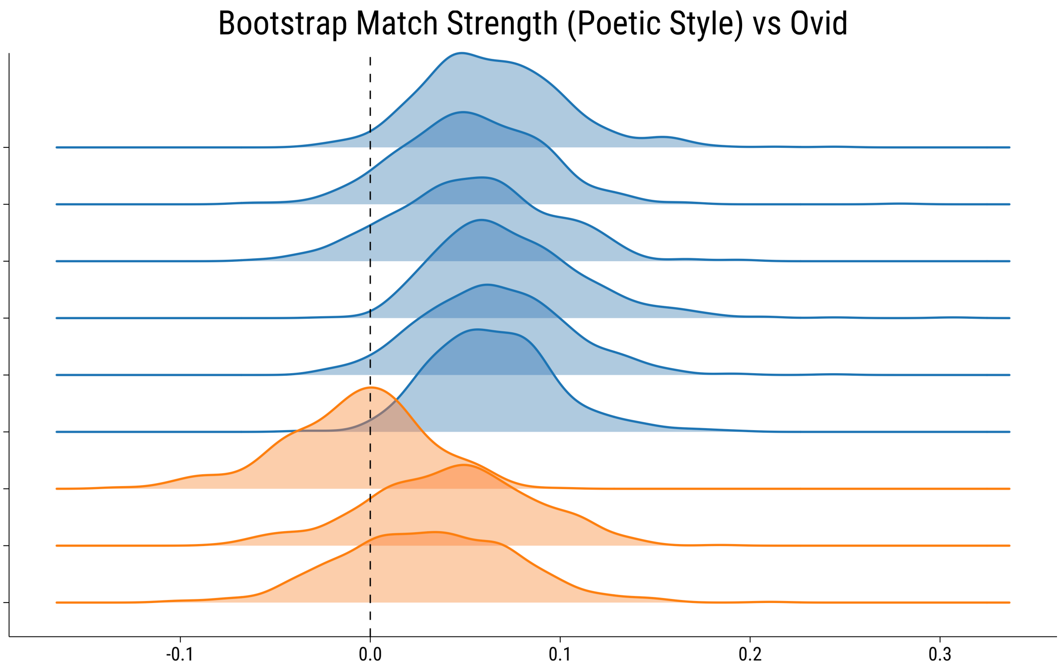


Figure 4: A BDI comparison of the same works attributed to Ovid, examining poetic/metrical features of Latin dactylic elegy instead of lexical features.

metrical technique was a powerful enough stylistic feature to disambiguate even deliberate imitation from genuine works. In Figure 3, we see the value of visualising distributions where all of the distances are positive (closer to the candidate author than an imposter), which would be summarised as a 'probability' of 1.0. This figure measures similarity in terms of lexico-grammatical features, operationalized as character *n*-grams. In fact, as can be seen, although the chunks from the *Consolatio* are much more like Ovid than they are like any of the distractor poets, they are *not as much like* Ovid as most of the

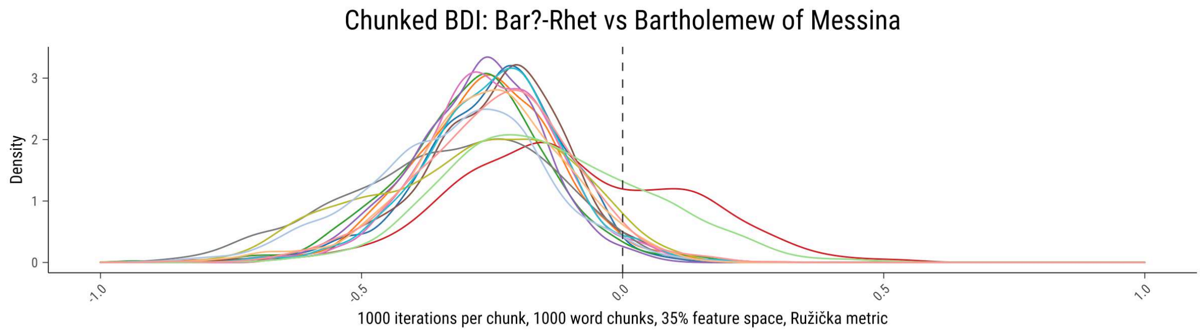


Figure 5: A BDI comparison of use of Latin function words to match a translation of Aristotle's *Rhetoric* to Bartholemew of Messina. Each distribution is the full BDI run for one chunk of the work.

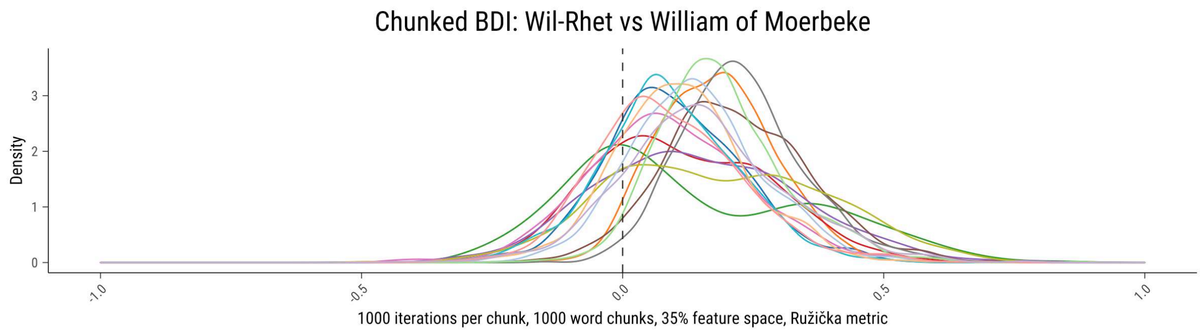


Figure 6: A BDI comparison of use of Latin function words to match a translation of Aristotle's *Rhetoric* to William of Moerbeke. Each distribution is the full BDI run for one chunk of the work.

candidate comparison works. This kind of comparability between strong matches is very difficult with the standard GI approach. However, in Figure 4, which measures metrical features, the difference is clear—the sections from the *Consolatio* are centered around 0 (or near enough) as compared to the other works where at least 90% of the distribution mass is above 0, supporting a positive attribution. This result suggests that the *Consolatio* is not Ovidian, but also that it is not a good stylistic match for any of the distractor poets (Tibullus, Propertius, and Catullan elegy). This is consistent with the current (weak) consensus that the *Consolatio* is a late first-century imitation by an unknown.

Figures 5 & 6 are from an analysis of translator style, examining medieval translations from Greek to Latin. Translator style (as opposed to authorial style) offers a unique set of challenges, covered in much more detail in the full paper [1]. In this study, a small set of function words was used, in accordance with the well-known theory that closed-class words are used more unconsciously, and thus are more indicative of individual preferences than nouns, verbs, and adjectives (which are highly affected by genre and topic). Overall, this was found to be an effective approach. Here, however, I note two more useful properties of the BDI method. The first is that by splitting the work into smaller chunks, and visualising the distribution for each chunk we are able to see the degree of stylistic variation in a single translator. It is also clear from Figure 5 that some passages are less 'stylistically clear', showing much more pronounced spread—this can be interpreted as greater sensitivity to the individual feature subsets. Overall, Figure 5 is centred around a negative value, indicating that it is significantly more similar to one of the imposter translators than to Bartholemew. In Figure 6, we performed the same process for a different text that is a translation of the same work (Aristotle's *Rhetoric*) generally accepted to be by William of Moerbeke. In the latter case we see the expected result—almost all of the chunks are fairly strongly centred around a positive value. The strength of the match is not as clear as in the Ovidian figures, but this is perhaps to be expected, since the amount of style that a translator brings to a work can be reasonably assumed to be less than that brought by an author (this is a well studied field; see for example [15] with references).

5. Future Work

As can be seen from Figure 2, predictions from the BDI classifier (after shifting) cluster strongly at the extremes, with most mis-predictions being high-confidence false n egatives. The Kestemont GI classifier shows the most mis-predictions in the central band (near 0.5), which is intuitive if the outputs are interpreted as probabilities. The current link function from the BDI distributions to a 'probability' in $[0, 1]$ is a fairly simple idea, and can almost certainly be improved to produce a smoother and more statistically informed distribution across the output range (perhaps logistic regression, or even empirical distributional statistics). This is left for future work.

6. Conclusions

The most common goal in authorship verification work is to positively attribute works to a uthors. In this context, although balanced accuracy is not unimportant, precision (fewer false positives) is often more important than recall. The balanced metrics used for the PAN 2014 / 2021 authorship verification competitions balance overall AUC (false positives and false negatives) with the ability for a classifier to degrade gracefully when the result is unclear. In general, this is a useful innovation, particularly in comparison to standard machine-learning classifiers which are obliged to assign each problem to a discrete class (even if the true author is not one of the available answers). While the widely-used General Imposters method still performs extremely well, it seems wasteful to discard the detailed distance information that is calculated in any case during the bootstrap / voting process.

BDI attempts to address these issues by outputting a full distance distribution which can be manually inspected. As demonstrated in Section 4, this can be very useful when comparing results that are all strong matches. When operating as a summary verifier, BDI tends to be conservative in its positive attributions, particularly when applied to very difficult problem sets like the PAN2014 *e n_novels*. In terms of raw performance, the BDI verifier appears slightly stronger than the improved Kestemont GI according to the PAN metrics for both the 2014 and 2021 problems, while also offering superior interpretability. The advantage of the BDI verifier is even clearer when score shifting is not us ed. Overall, the BDI approach seems to be a strong choice, especially where training data is limited and/or reliable positive results are more important than balanced performance metrics.

7. Availability of Data and Code

The preprint may be found at https://github.com/bnagy/bdi-paper. All code and data is available under CC-BY, except where restricted by upstream licenses. The code repository includes full reproduction data and code for the evaluation, as well as various supplemental figures and explanations.

Acknowledgments

This work was supported by project 2020/39/O/HS2/02931, funded by Poland's National Science Centre

References

[1] Pieter Beullens, Wouter Haverals, and Ben Nagy. "The Elementary Particles: A Computational Stylometric Inquiry into the Mediaeval Greek-Latin Aristotle". In: *Mediterranea. International Journal on the Transfer of Knowledge* 9 (2024), pp. 385–408. DOI: 10.21071/mijtk.v9i.16723.

[2] Janek Bevendorff et al. "Generalizing Unmasking for Short Texts". In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Ed. by Jill Burstein, Christy Doran, and Thamar Solorio. Minneapolis, Minnesota: Association for Computational Linguistics, 2019, pp. 654–659. DOI: 10.18653/v1/N19-1068.

[3] Benedikt Boenninghoff, Robert M. Nickel, and Dorothea Kolossa. "O2D2: Out-Of-Distribution Detector to Capture Undecidable Trials in Authorship Verification—Notebook for PAN at CLEF 2021". In: *CLEF 2021 Labs and Workshops, Notebook Papers*. Ed. by Guglielmo Faggioli et al. CEUR-WS.org, 2021. URL: http://ceur-ws.org/Vol-2936/paper-158.pdf.

[4] Maciej Eder, Jan Rybicki, and Mike Kestemont. "Stylometry with R: a package for computational text analysis". In: *R Journal* 8.1 (2016), pp. 107–121. URL: https://journal.r-project.org/archive/2016/RJ-2016-007/index.html.

[5] Mike Kestemont. *Ružička: Authorship Verification in Python*. 2015. URL: https://github.com/mikekestemont/ruzicka.

[6] Mike Kestemont et al. "Authenticating the writings of Julius Caesar". In: *Expert Systems with Applications* 63 (2016), pp. 86–96. DOI: https://doi.org/10.1016/j.eswa.2016.06.029.

[7] Mike Kestemont et al. "Overview of the Cross-Domain Authorship Verification Task at PAN 2021". In: *CLEF 2021 Workshop Proceedings*. CEUR-WS.org, 2021. URL: https://ceur-ws.org/Vol-2936/paper-147.pdf.

[8] Mahmoud Khonji and Youssef Iraqi. "A slightly-modified GI-based author-verifier with lots of features (ASGALF) - Notebook for PAN at CLEF 2014". In: *CLEF 2014 Working Notes* 1180 (2014), pp. 977–983. URL: https://ceur-ws.org/Vol-1180/CLEF2014wn-Pan-KonijEt2014.pdf.

[9] Moshe Koppel and Yaron Winter. "Determining If Two Documents Are Written by the Same Author". In: *Journal of the Association for Information Science and Technology* 65 (2014). DOI: 10.1002/asi.22954.

[10] Ben Nagy. *Preprint: Bootstrap Distance Imposters: High precision authorship verification with improved interpretability*. 2024. URL: https://github.com/bnagy/bdi-paper.

[11] Ben Nagy. *Ružička: Authorship Verification in Python*. 2023. URL: https://github.com/bnagy/ruzicka.

[12] Andrea Nini. *A Theory of Linguistic Individuality for Authorship Analysis*. Elements in Forensic Linguistics. Cambridge University Press, 2023. DOI: 10.1017/9781108974851.

[13] F. Pedregosa et al. "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[14] Nektaria Potha and Efstathios Stamatatos. "An Improved Impostors Method for Authorship Verification". In: *Experimental IR Meets Multilinguality, Multimodality, and Interaction: 8th International Conference of the CLEF Association, CLEF 2017, Dublin, Ireland, September 11–14, 2017, Proceedings 8*. 2017, pp. 138–144. DOI: 10.1007/978-3-319-65813-1_14.

[15] Jan Rybicki. "The great mystery of the (almost) invisible translator". In: *Quantitative Methods in Corpus-Based Translation Studies: A practical guide to descriptive translation research*. Ed. by Michael P. Oakes and Meng Ji. John Benjamins Publishing Company Amsterdam, 2012, pp. 231–248.

[16] Efstathios Stamatatos et al. "Overview of the author identification task at PAN 2014". In: *CEUR Workshop Proceedings* 1180 (2014), pp. 877–897.

[17] Janith Weerasinghe, Rhia Singh, and Rachel Greenstadt. "Feature Vector Difference based Authorship Verification for Open-World Settings." In: *CLEF 2021 Labs and Workshops, Notebook Papers*. Ed. by Guglielmo Faggioli et al. CEUR-WS.org, 2021. URL: https://ceur-ws.org/Vol-2936/paper-197.pdf.