Bootstrap Distance Imposters: A less accurate classifier for a more civilized age

Ben Nagy[1]

[1]*Institute of Polish Language, Polish Academy of Sciences (IJP PAN)*
Adama Mickiewicz 31
Kraków, Poland

Abstract

This paper describes an update to Mike Kestemont and Walter Daeleman's open-source Python implementation of the General Imposters method of authorship attribution. The new algorithm, called Bootstrap Distance Imposters (henceforth BDI), incorporates several of the improvements proposed since the code was last updated in 2015, 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 [1], and some additional properties of BDI are showcased via real-world case studies. BDI is shown to be a high-precision (few false positives) classifier for authorship attribution, with competitive overall accuracy, and significant advantages in interpretability.

Keywords

authorship attribution, stylometry, bootstrapping,

1. Introduction

The General Imposters method (henceforth GI), originally formulated by M. Koppel and Y. Winter in [2], has become one of the standard methods for authorship attribution. After strong performances in the PAN 2013 and 2014 competitions, it was implemented and used by M. Kestemont et al. in an influential study [3], improved by N. Potha and E. Stamatatos in 2017 [4], and is now available in the well-known R package *stylo* [5].

In this paper I describe an update to Kestemont's open-source Python implementation [6] 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 [1], and some additional properties are showcased via real-world case studies.

2. Motivation and Design

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 as a probability. 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 $d : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$. If the candidates are further, the difference between the distances is negative, if closer it is positive. If these individual distances follow a

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

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 3 below). The 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 features). 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 M. Khonji and Y. Iraqi [7] also modified the classic GI algorithm to incorporate the distance between vectors (in that case the relative distance of the test vector to candidates vs imposters is used in the decision function for a 'standard' voting-based classifier), so this paper is not the first to note the utility of this additional information.

2.1. Classification Performance

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 attribution 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 that an imposter then the result will be 1, if every test is more like an imposter, it will be 0, etc. This is implemented as `(100-scipy.stats.percentileofscore(x, 0)) / 100.0`, 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 `sklearn` [8] conventions like `fit()` and `predict_proba()` and evaluated the BDI classifier directly against the updated Kestemont GI `Order2Verifier`, using the 2014 PAN authorship attribution problems. This provided a convenient benchmark, and also the opportunity to compare the results against a number of other (although older) verification approaches.

2.2. Score Shifting

The PAN 2014 competition provided a set of training problems, and the c@1 metric introduced in that competition 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 0.5, hopefully improving the c@1 score as compared to basic accuracy.

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))
```

[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.

```
        else:
            new_scores.append(0.5)
```

Scores below p1 are scaled to $[0, \texttt{p1})$, scores above p2 are scaled to $(\texttt{p1}, 1]$, 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 in Kestemont).

```
    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 based on first principles.

2.3. Changes to Kestemont GI

As is the nature of computing, 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 [9].

- 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 [10]
- Implement the Potha & Stamatatos 'ranking' improvement for the consensus score[2] described in [4]
- Remove most non-core code
- Modify the score-shifting algorithm, as described above

2.4. Testing

The BDI classifier was compared head-to-head with the updated Kestemont `Order2Verifier` on the full PAN 2014 evaluation corpus, as archived in the github repository.[3] For this test I used two different

[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 (with enough bootstrapping this should converge)

[3]There is a small inconsistency that I was unable to resolve—the verification problems stored at the Kestemont repository 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

Classifier	Vectorizer	Shifter	Accuracy	AUC	C@1	Final Score
BDI, Cosine	2,3,4,5-grams	fitted	0.681	0.727	0.694	0.505
		manual	0.672	0.715	0.649	0.464
	2,3,4-grams	fitted	0.686	0.723	0.689	0.499
		manual	0.681	0.715	0.645	0.461
BDI, Minmax	2,3,4,5-grams	fitted	**0.689**	0.731	0.695	0.508
		manual	0.682	0.726	0.660	0.479
	2,3,4-grams	fitted	0.682	0.730	0.694	0.507
		manual	0.685	0.723	0.652	0.472
Kestemont GI, Cosine	2,3,4,5-grams	fitted	0.673	0.768	0.695	0.534
		manual	0.621	0.698	0.557	0.389
	2,3,4-grams	fitted	0.665	0.755	0.678	0.512
		manual	0.614	0.685	0.537	0.367
Kestemont GI, Minmax	2,3,4,5-grams	fitted	0.661	**0.773**	**0.702**	**0.543**
		manual	0.633	0.706	0.568	0.401
	2,3,4-grams	fitted	0.668	0.759	0.694	0.527
		manual	0.629	0.707	0.572	0.405
PAN 2014 Best (individual)			NA	0.718	0.684	0.490

Table 2
BDI 2,3,4,5-grams, Minmax, Manual Shifter

Corpus	Tests	??	High Conf.	FP	Acc .	AUC	C@1	Final	PAN Best
du_essays	96	19	75	2	0.854	0.955	0.923	**0.882**	0.823
du_reviews	50	10	35	0	0.580	0.693	0.576	0.399	**0.525**
en_essays	200	27	162	0	0.605	0.624	0.568	0.354	**0.513**
en_novels	200	34	161	0	0.595	0.622	0.556	0.346	**0.508**
gr_articles	100	34	61	2	0.800	0.848	0.697	0.591	**0.720**
sp_articles	100	35	53	1	0.780	0.882	0.810	**0.715**	0.698

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

Corpus	Tests	??	High Conf.	FP	Acc .	AUC	C@1	Final	PAN Best
du_essays	96	16	54	2	0.854	0.964	0.948	**0.914**	0.823
du_reviews	50	12	14	7	0.640	0.696	0.645	0.449	**0.525**
en_essays	200	14	22	73	0.565	0.569	0.578	0.329	**0.513**
en_novels	200	42	79	12	0.610	0.670	0.629	0.421	**0.508**
gr_articles	100	28	18	3	0.790	0.840	0.755	0.634	**0.720**
sp_articles	100	15	27	6	0.750	0.935	0.863	**0.807**	0.698

sets of character n-grams, since that feature universe is a generally reliable and uncontroversial choice for modern authorship attribution 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 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, with fitted and manual score shifting. 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 in his 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 [11].

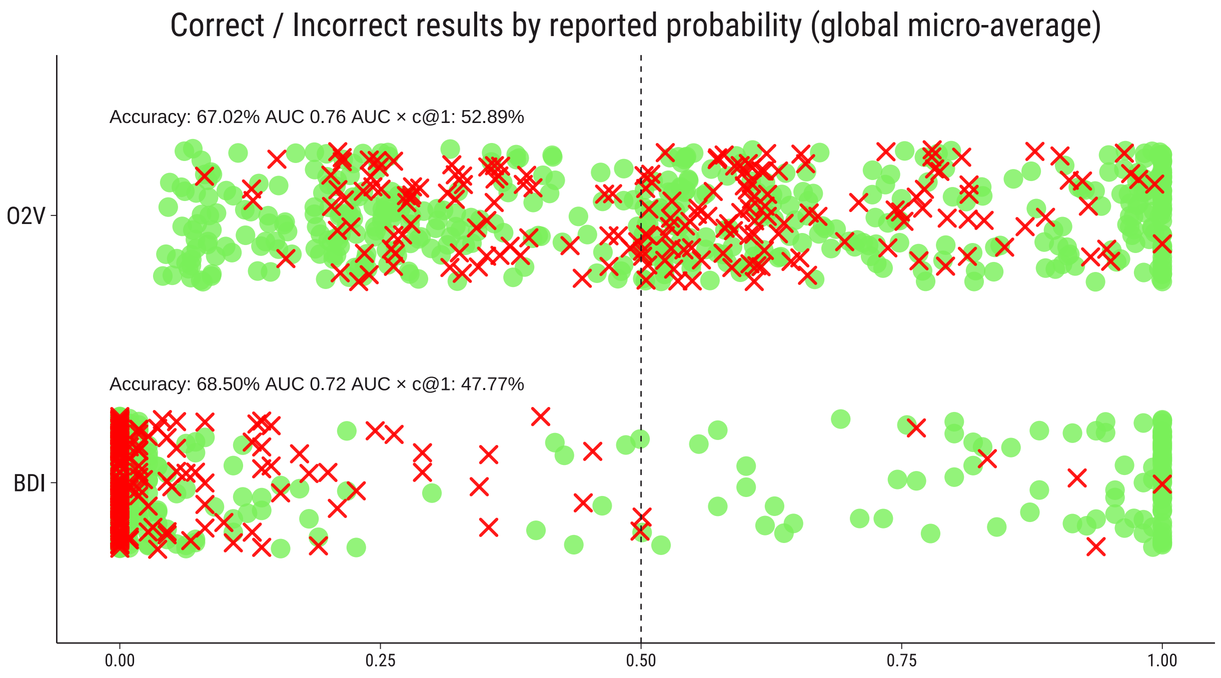

Figure 1: A comparison of the correct and incorrect results by reported probability for BDI (manual shifter, 2,3,4,5-grams, minmax) vs Order2Verifer (fitted shifter, 2,3,4,5-grams, minmax)

2.5. Interpretation

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. The GI `Order2Verifier` with a fitted shifter performs better according to the PAN metrics (deriving a large c@1 benefit from fitting the score shifter). The GI classifier also appears to outperform the best overall PAN 2014 entrant by a significant margin. 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 classifier whose positive results are reliable (at the cost of more false negatives). This can be seen clearly in Figure 1 in which the best performing GI classifier is compared to the manually fitted BDI classifier (results between 11% and 89% are left unanswered), using the same features and metrics. The results for each subcorpus are broken down in more detail in Tables 2 & 3.

3. Showcase

In this section I refer to two attribution studies I have contributed to that are, 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 2 & 3 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 metrical technique was a powerful enough stylistic feature to disambiguate even deliberate imitation from genuine works. In Figure 2, 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

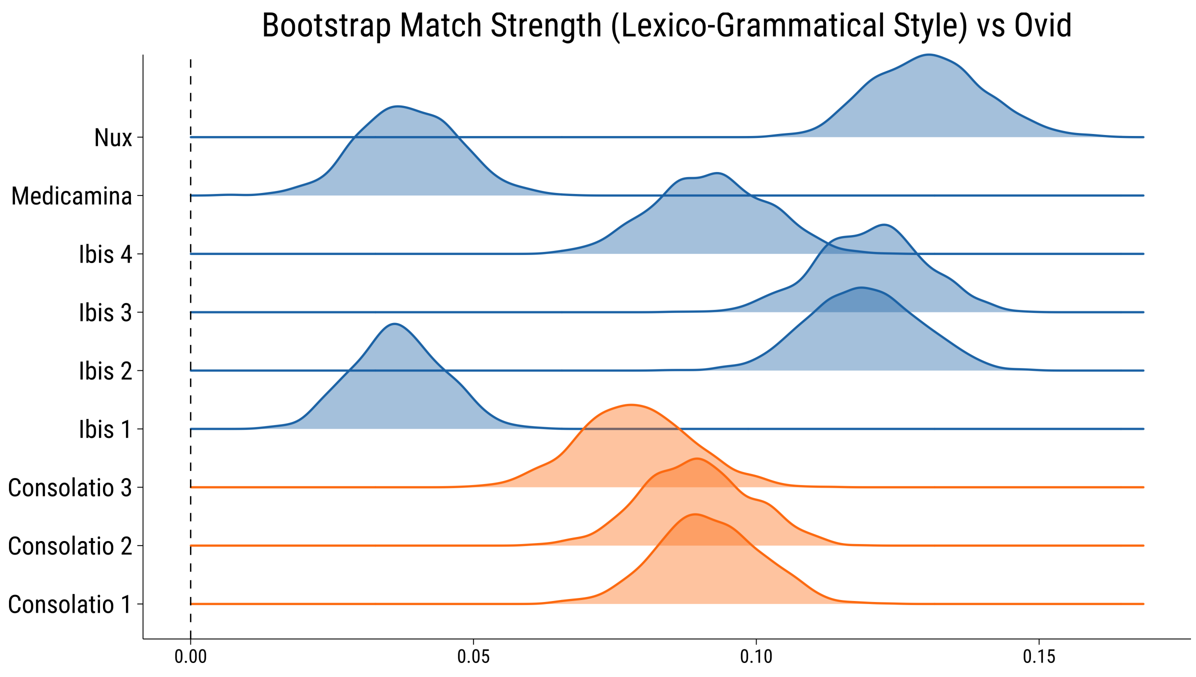

Figure 2: A BDI comparison of several works attributed to Ovid using lexico-grammatical features (character n-grams). The *Consolatio Ad Liviam* is now considered to be a first-century interpolation.

like Ovid than they are like any of the distractor poets, they are *not as much like* Ovid as most of the comparison works. This kind of comparability between strong matches is very difficult with the standard GI approach. However, in Figure 3 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, representing 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 Catullus).

Figures 4 & 5 are from an analysis of translator style, examining medieval translations from Greek to Latin. 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 4 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 4 is centred around a negative value, indicating that it is significantly more similar to one of the imposter translators than to Bartholemew. In Figure 5, we performed the same process for a different text that is a translation of the same work (Aristotle's *Rhetorics*) generally accepted to be by William of Moerbeke. In this 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.

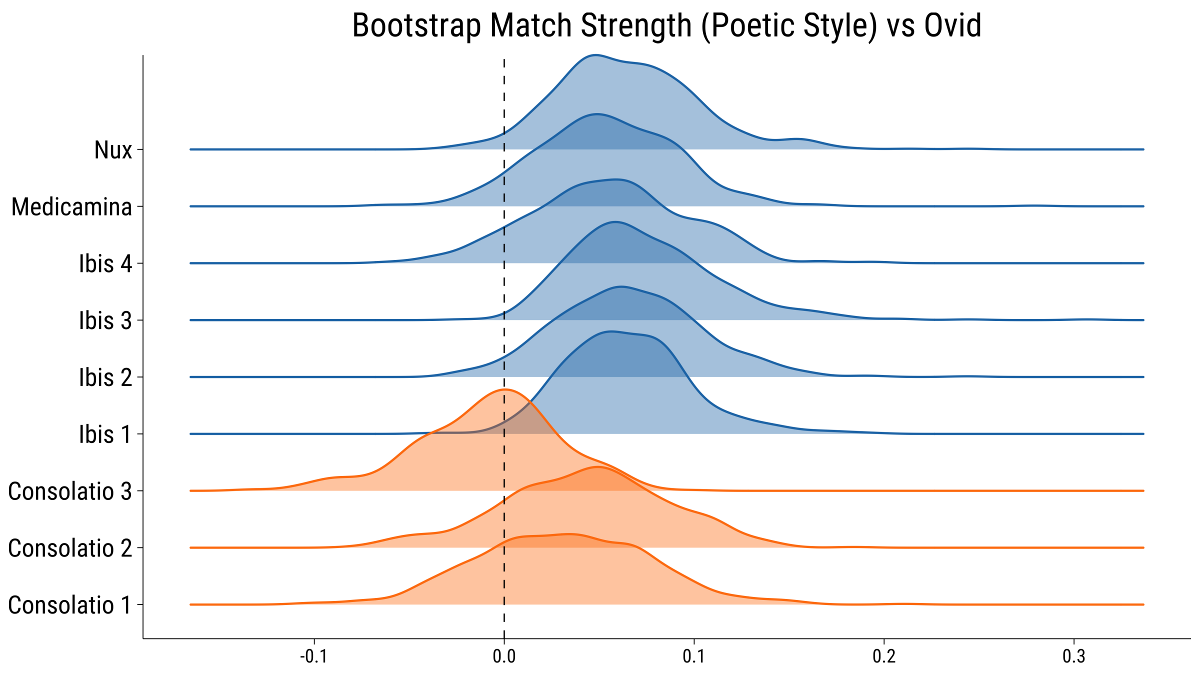

Figure 3: A BDI comparison of the same works attributed to Ovid, examining poetic/metrical features of Latin dactylic elegy instead of lexical features.

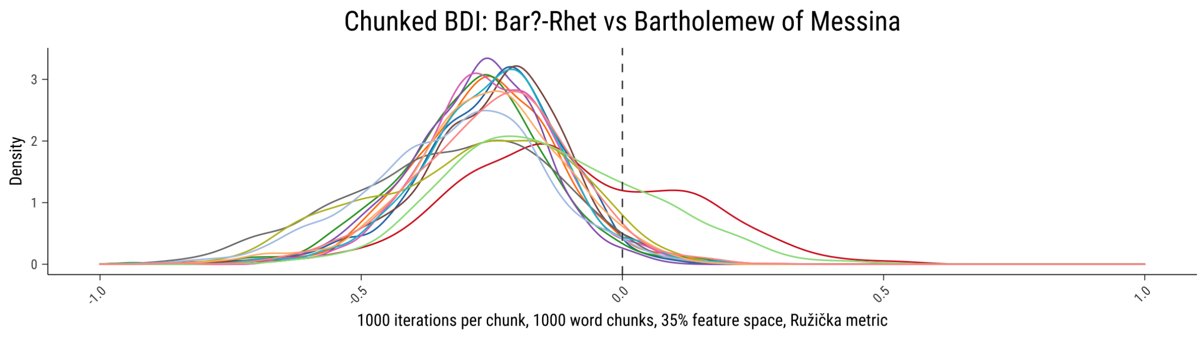

Figure 4: A BDI comparison of use of Latin function words to match a translation of Aristotle's *Rhetorics* to Bartholemew of Messina. Each distribution is the full BDI run for one chunk of the work.

4. Future Work

As can be seen from Figure 1, predictions from the BDI classifier (after shifting) cluster strongly at the extremes, with most mis-predictions being high-confidence false negatives. 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 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 believable distribution across the output range (perhaps logistic regression, or even empirical distributional statistics). This is left for future work.

5. Conclusions

The most common task in authorship attribution work is to positively attribute works to authors. In this context, although balanced accuracy is not unimportant, precision (fewer false positives) is often more

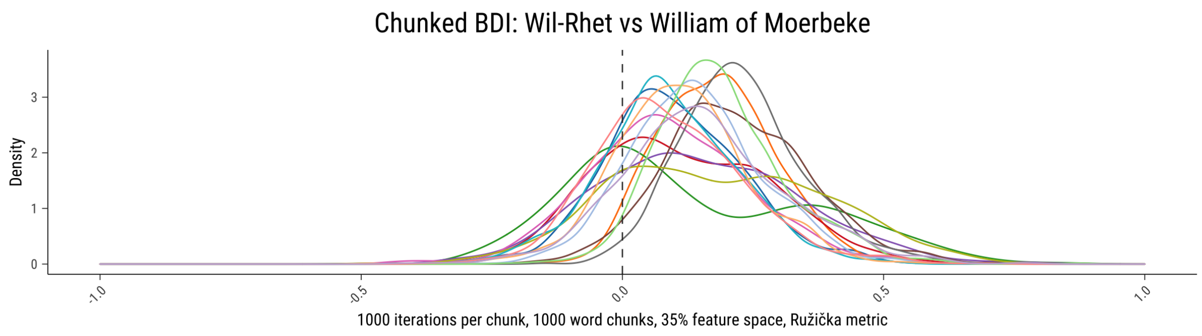

Figure 5: A BDI comparison of use of Latin function words to match a translation of Aristotle's *Rhetorics* to William of Moerbeke. Each distribution is the full BDI run for one chunk of the work.

important than recall. The c@1 metric introduced for the PAN 2014 authorship verification competition balances overall AUC (false positives and false negatives) with the ability for a classifier to decline to answer problems 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 GI method 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 3, this can be very useful when comparing results that are all strong matches. When operating as a summary classifier, BDI tends to be conservative in its positive attributions, showing a false positive rate on the PAN 2014 test problems of less than 1% when operating with a score shifter that demanded an 89% confidence match. While the raw c@1 results of the BDI vectorizer/classifier/shifter combinations were not as strong as the best result from updated Kestemont GI, the fitted versions all outperformed the best PAN 2014 entry, and the more conservative versions (with manual shifting) would all have finished in the top three. 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 AUC $\times$ c@1 performance.

6. 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 Polish Academy of Sciences Grant 2020/39 / O / HS2 / 02931.

References

[1] E. Stamatatos, W. Daelemans, B. Verhoeven, M. Potthast, B. Stein, P. Juola, M. Sanchez-Perez, A. Barrón-Cedeño, Overview of the author identification task at PAN 2014, CEUR Workshop Proceedings 1180 (2014) 877–897.

[2] M. Koppel, Y. Winter, Determining if two documents are written by the same author, Journal of the Association for Information Science and Technology 65 (2014). doi:10.1002/asi.22954.

[3] M. Kestemont, J. Stover, M. Koppel, F. Karsdorp, W. Daelemans, Authenticating the writings of Julius Caesar, Expert Systems with Applications 63 (2016) 86–96. URL: https://www.sciencedirect.com/science/article/pii/S0957417416303116. doi:https://doi.org/10.1016/j.eswa.2016.06.029.

[4] N. Potha, E. 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.

[5] M. Eder, J. Rybicki, M. Kestemont, Stylometry with R: a package for computational text analysis, R Journal 8 (2016) 107–121. URL: https://journal.r-project.org/archive/2016/RJ-2016-007/index.html.

[6] M. Kestemont, Ružička: Authorship verification in Python, 2015. URL: https://github.com/mikekestemont/ruzicka.

[7] M. Khonji, Y. Iraqi, A slightly-modified GI-based author-verifier with lots of features (ASGALF) - notebook for PAN at CLEF 2014, CLEF 2014 Working Notes 1180 (2014) 977–983. URL: https://ceur-ws.org/Vol-1180/CLEF2014wn-Pan-KonijEt2014.pdf.

[8] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, Scikit-learn: Machine learning in Python, Journal of Machine Learning Research 12 (2011) 2825–2830.

[9] B. Nagy, Ružička: Authorship verification in Python, 2023. URL: https://github.com/bnagy/ruzicka.

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

[11] B. Nagy, Preprint: Bootstrap Distance Imposters: A less accurate classifier for a more civilized age (2024). URL: https://github.com/bnagy/bdi-paper.