Dynamic Ensemble Selection for Author Verification

Nektaria Potha and Efstathios Stamatatos

University of the Aegean
83200 Karlovassi, Greece
{nekpotha,stamatatos}@aegean.gr

Abstract. Author verification is a fundamental task in authorship analysis and associated with significant applications in humanities, cyber-security, and social media analytics. In some of the relevant studies, there is evidence that heterogeneous ensembles can provide very reliable solutions, better than any individual verification model. However, there is no systematic study of examining the application of ensemble methods in this task. In this paper, we start from a large set of base verification models covering the main paradigms in this area and study how they can be combined to build an accurate ensemble. We propose a simple stacking ensemble as well as a dynamic ensemble selection approach that can use the most reliable base models for each verification case separately. The experimental results in ten benchmark corpora covering multiple languages and genres verify the suitability of ensembles for this task and demonstrate the effectiveness of our method, in some cases improving the best reported results by more than 10%.

Keywords: Author verification, Authorship analysis, Ensemble learning, Dynamic ensemble selection

1 Introduction

Authorship analysis is a research area in text mining that attempts to reveal information about the authors of electronic documents. Among authorship analysis tasks, author verification is considered to be fundamental [19] since it focuses on the most basic question: whether two documents are written by the same author. More complex tasks like authorship attribution (i.e., identifying the most likely author given a closed-set or open-set of suspects) [31] or authorship clustering (grouping a collection of documents by authorship) can be decomposed into a series of author verification cases [21].

Technology in author verification is strongly associated with applications in several fields. In digital humanities, author verification can be used to reveal the identity of authors of documents of high historical and literary importance [35, 36]. In cyber-security, it can be used to detect compromised accounts [3] or spearphishing attacks [8] and enable continuous authentication of users [5]. Author verification can also be used to detect multiple accounts controlled by the same user [1] and facilitate deception detection [22] in social media.

A typical author verification case (or instance) is a tuple $(D_{known}, d_{unknown})$ where D_{known} is a set of documents of known authorship, all by the same author, and $d_{unknown}$ is another document of questioned authorship. An author verification method should be able to decide whether or not the author of D_{known} is also the author of $d_{unknown}$. Apart from a binary (yes/no) answer, author verification methods usually produce a verification score in [0,1] that can be viewed as a confidence estimation [33, 34]. Essentially, author verification is a one-class classification task since only labelled samples from the positive class are available [10]. However, there are approaches that attempt to transform it to a binary classification task by sampling the negative class (i.e., all documents by all other authors) [21].

Recently, several methods have been proposed in the relevant literature [32], largely motivated by the corresponding PAN shared tasks organized from 2013 to 2015 [14, 33, 34]. In some previous works, there is evidence that an ensemble of verifiers could be better than any single model. The organizers of PAN used a simple heterogeneous ensemble by averaging all verification scores produced by the submitted methods and found that this simple meta-model was far better than any individual verifier in PAN-2014 [34]. A similar attempt in PAN-2015 shared task did not provide equally impressive results, mainly due to the very low performance of many submissions in that case [33]. However, another heterogeneous ensemble combining five verifiers won the second-best overall rank in PAN-2015 [23]. So far, there is lack of more systematic studies examining a large pool of verifiers and more sophisticated ensemble learning approaches.

In the current paper, we attempt to fill that gap by starting from a wide range of base verification models covering the most important paradigms in the relevant literature. Then, we propose two ensemble learning approaches. First a simple stacking method using a meta-learner to combine the outputs of base models. Second, a dynamic ensemble selection method that can focus on the most effective models for each verification case separately. Experimental results on several benchmark datasets covering different languages and genres demonstrate the effectiveness of both approaches especially in challenging cases where D_{known} is of limited size and in cross-domain conditions.

The rest of this paper is organized as follows: Section 2 presents previous work in author verification while Section 3 describes our proposed methods. The performed experiments are analytically presented in Section 4 while Section 5 discusses the main conclusions and suggests future work directions.

2 Previous Work

Early work in this field is marked by the *unmasking* approach [20] that builds a classifier to distinguish between two documents and examines how fast the accuracy drops when the most important features are gradually removed. This method was found to be very effective in long literary documents but not reliable when only short text samples are available [29] or when cross-genre conditions are met [15]. Research in author verification has been strongly influenced by

the recent PAN shared tasks [14, 33, 34] where multiple submitted methods were evaluated in several benchmark datasets covering different languages and genres.

In general, author verification methods follow specific paradigms [32]. First, *intrinsic* methods attempt to handle a one-class classification task by initially estimating the similarity of $d_{unknown}$ to D_{known} and then deciding whether this similarity is significant [10, 13, 18, 25]. Usually, such approaches are fast and robust across different domains and languages.

On the other hand, *extrinsic* methods attempt to transform author verification to a binary classification task by sampling the negative class which is huge and extremely heterogeneous (it comprises all other possible authors) [16, 21, 26, 30]. Then, extrinsic methods attempt to decide if the similarity of $d_{unknown}$ to D_{known} is higher than the similarity of $d_{unknown}$ to $D_{external}$ (the collected samples of the negative class). All top-ranked submissions to PAN shared tasks from 2013 to 2015 follow this paradigm [2, 16, 30] demonstrating its effectiveness. However, the performance of such methods heavily depends on the quality of the collected external documents [21].

From another point of view, author verification methods can be distinguished according to the way they handle the members of D_{known}. The *instance-based* approaches [31] treat each known document separately and then combine the corresponding decisions [6, 16, 30]. If there is only one known document, some instance-based methods segment it into parts to enable the estimation of variance of similarity within D_{known} [13]. On the contrary, *profile-based* techniques [31] concatenate all known documents attempting to better represent the properties of the style of author, rather than the style of each document [10, 18, 25]. This approach is better able to handle short texts in comparison to instance-based methods. However, it disregards any useful information about the variation it might exist within the set of known documents.

Another category of methods focus on the representation of verification instances (i.e., the tuple (D_{known},$d_{unknown}$)) rather than the individual documents they contain. Each verification instance may be positive (same author) or negative (different author) and given a training dataset of such instances a classifier is build to learn to distinguish between these two classes [4, 9, 12]. Such *eager* approaches [32] attempt to learn a general verification model and its effectiveness strongly depends on the volume, representativeness and distribution of the training dataset [33].

Regarding the stylometric information extracted from documents, most author verification approaches are based on simple but effective features like word and character n-grams [13, 18, 21, 25]. There are also language-agnostic approaches using information extracted from text compression [10]. The use of NLP tools to extract syntactic-related information is limited [4, 23]. A recent study based on representation learning uses a neural network to jointly learn character, lexical, topical, and syntactic modalities and achieved very good results [7]. Another deep learning method, the winning approach in PAN-2015, uses a character-level recurrent neural network language model [2]. In addition, recently, the use of topic modeling techniques provided promising results[11, 27].

Table 1. Distribution of base verification models over the different paradigms.

	Intrinsic	Extrinsic
Instance-based	16	10
Profile-based	16	5

3 The Proposed Methods

3.1 Base Verification Models

In this paper, we use an extended list of author verification models that cover the main paradigms in this area. More, specifically, we implemented 47 base models that belong to the following categories:

- *Instance-based Intrinsic Models*: These are inspired from [13], a very robust method that was used as baseline in PAN-2014 and PAN-2015 shared tasks [33, 34].
- *Profile-based Intrinsic Models*: We adopt a simple but effective method described in [27].
- *Instance-based Extrinsic Models*: The well-known General Impostors (GI) method [30] as well as a recently-proposed modification called ranking-based impostors [26] are used.
- *Profile-based Extrinsic Models*: We use another modification of GI that follows the profile-based paradigm [28]. For each verification instance, all known documents (D_{known}) are first concatenated. Then, multiple artificial impostor documents of similar properties are formed by concatenating an equal number ($|D_{known}|$) of external documents.

The variation of models in each category consists of using different text representation schemes. Several feature types (word unigrams, character 3-grams, 4-grams, or 5-grams) are used and two topic modeling techniques (Latent Semantic Indexing (LSI) or Latent Dirichlet Allocation (LDA)) are applied to build various version of a certain verifier. We also examine two different corpora (a small and a larger one) to extract the topic models. More details on how the base models are used and tuned in the performed experiments are given in Section 4.2. The distribution of base verification models is shown in Table 1. Extrinsic models are fewer than intrinsic ones because less variation in feature types is used in that case. That way the distribution of our base verifiers is similar to those of methods submitted to PAN shared tasks [32], where intrinsic methods were more popular than extrinsic ones while the majority of submissions followed the instance-based paradigm.

input : v, T, M, a, b
output: $fusedScore$
foreach $t \in T$ **do**
  if $cosine(vector(v), vector(t)) \geq a$ **then**
    $T_{similar} = T_{similar} \cup t$
  end
end
foreach $m \in M$ **do**
  $weight(m) = accuracy(m, T_{similar})$
  if $weight(m) \geq b$ **then**
    $M_{suitable} = M_{suitable} \cup m$
  end
end
$fusedScore = \sum_{m \in M_{suitable}} weight(m) \cdot score(m, v)$

Algorithm 1: The proposed DES author verification method.

3.2 Stacking Ensemble

First, we focus on the use of a simple method to construct heterogeneous ensembles. A meta-learner (binary classifier) can be trained based on the output of the 47 base verifiers following a well-known *stacked generalization* approach [37]. Such a model can learn the correlations between the input features and the correctness of base models. After performing some preliminary experiments examining several alternative classifiers (e.g. multi-layer perceptron, k-nn), we finally use a Support Vector Machine (SVM) meta-learner in our stacking ensemble since in most of the cases it provides the most competitive results.

It has to be noted that the performance of such an approach heavily depends on the distribution of verification instances over the two classes (same author or different author). As it is explained in Section 4.1, all datasets we use in this study are balanced. However, in case the distribution of instances over the classes is not balanced, or not known, then a more carefully selected meta-learner (able to handle the class imbalance problem) should probably be used.

3.3 Dynamic Ensemble Selection

In this paper, we also propose a more sophisticated ensemble that is based on Dynamic Ensemble Selection (DES) [17] that focuses on suitable base models for each verification instance separately. Our method requires a training dataset, i.e. a collection of verification instances in the same language and genre with respect to the test dataset. In particular, given a test verification instance v, a collection of training instances T, and a collection of base verification models M (each model can produce a score in [0,1] when it gets as input a verification instance), our DES method performs the following steps (see also algorithm 1):

1. Represent the characteristics of each (training or test) verification instance $(D_{known}, d_{unknown})$ as a numerical vector reflecting how homogeneous known

Table 2. The similarity features used to represent each (training or test) verification instance ($D_{known}, d_{unknown}$). An additional feature is the size of D_{known}, so the total number of features is 73.

Similarity	Function	Fusion	Representation	#Features
D_{known} vs. $d_{unknown}$, or within D_{known}	Cosine, Minmax, or Euclidean	min, max, or avg	word unigrams, char 3-grams, char 4-grams, or char 5-grams	$2 \times 3 \times 3 \times 4 = 72$

documents are and how distant they are from the unknown document. First, each (known or unknown) document is represented based on a certain feature type (word or character n-grams) and then similarity between documents is calculated. We focus on two types of similarity. First, within members of D_{known} that shows the degree of homogeneity in the known document set. Second, we compare all members of D_{known} to the unknown document to estimate how close they are. Since D_{known} usually includes multiple documents, a fusion method is needed to combine the obtained similarity values for each known document. In more detail, we use 3 similarity functions, 3 fusion methods, and 4 text representation types (see Table 2) to calculate 72 similarity features for each verification instance. The final vector contains one more feature that corresponds to the size of D_{known}.

2. Calculate the similarity of the test instance vector to each of the training instance vectors using cosine similarity.
3. Filter out all training instances with similarity to the test instance lower than a threshold a. Let $T_{similar} \subset T$ be the set of the remaining training instances highly similar to the test instance.
4. Calculate the effectiveness of each base verification model (47 in total) on $T_{similar}$.
5. Filter out all base verification models with effectiveness on $T_{similar}$ lower than a threshold b. Let $M_{suitable} \subset M$ be the set of the selected verification models.
6. Apply the remaining base verification models ($M_{suitable}$) to the test instance.
7. Fuse the scores of $M_{suitable}$ on v according to a weighted average where the weight of each model is determined by its effectiveness on $T_{similar}$.

The proposed method has two important parameters, thresholds a and b. The former determines the size of the set of selected training instances. If it is set too high (e.g., 0.9), very few similar training instances would be found. The latter affects the number of selected base verification models. If it is set too high, very few base verification models will be considered to provide the final answer. It should also be noted that some of the features used to represent a verification instance become useless when there is only one known document. In more detail, when there is exactly one known document then the features that calculate the similarity within the set of known documents are all equal.

Table 3. The PAN benchmark datasets used in this study ($|d|$ denotes text length in words).

	Dataset	Training instances	Test instances	$avg(\|D_{known}\|)$	$avg(\|d\|)$
PAN-2014	DE (Dutch Essays)	96	96	1.89	405
	DR (Dutch Reviews)	100	100	1.02	114
	EE (English Essays)	200	200	2.62	841
	EN (English Novels)	100	200	1.00	5115
	GR (Greek Articles)	100	100	2.77	1470
	SP (Spanish Articles)	100	100	5.00	1129
PAN-2015	DU (Dutch Cross-genre)	100	165	1.75	357
	EN (English Cross-topic)	100	500	1.00	508
	GR (Greek Cross-topic)	100	100	2.87	717
	SP (Spanish Mixed)	100	100	4.00	950

4 Experiments

4.1 Description of Data

We consider benchmark corpora built in the relevant PAN evaluation campaigns on authorship verification in 2014 and 2015. These corpora cover four languages (Dutch, English, Greek, and Spanish) and several genres (newspaper articles, essays, reviews, literary texts etc.) [33, 34]. Each corpus is divided into a training and a test part and in each case multiple verification instances are provided. Each instance includes a small number (up to 10) of known documents, all by the same author, and exactly one questioned document(unknown document). It is noticeable each dataset, either training or test, is balanced with respect to the distribution of positive (same-author) and negative (different-author) instances.

In PAN-2014 datasets, all (known and unknown) documents within a verification instance share the same language, genre, and thematic area. On the other hand, PAN-2015 datasets are more challenging since they include cross-domain cases, i.e., all documents within a verification instance are in the same language but they may belong to distinct thematic areas or genres.

4.2 Setup

We follow the same evaluation procedure with PAN shared tasks to achieve compatibility of evaluation results with PAN participants [33, 34]. We use the training part of the each dataset to tune the parameters and calibrate the verification score of each model and then we apply the tuned models to the test part of the dataset. The parameter tuning is performed by grid search trying to optimize the Area Under the Receiver-Operating Characteristic Curve (AUC), an evaluation measure also used in PAN shared tasks. In addition, we perform five runs for the non-deterministic models (i.e., all variants of GI) and consider

their average verification score. The output of each base model for the training instances of a dataset is used to train a logistic regression classifier that can provide binary (same author or different author) answers.

Apart from the tuned models, we also examine base models with fixed parameter settings. The idea behind this is that DES is based on the performance of the (selected) base models on the (selected) training instances. Thus, if the base models are tuned based on the same dataset, their output on that specific dataset could be biased. Taking into account results of previous work [13, 26, 27, 30], we set each parameter of the base model to a default value (e.g., 250 latent topics when LSI or LDA is applied, 150 impostors per repetition in all variations of GI).

The set of external documents ($D_{external}$) used in the framework of extrinsic verification models is collected from the world wide web for each dataset separately following the procedure described in [26]. When topic modeling is applied, the latent topic models are extracted either from the documents in the training dataset exclusively or from a larger collection consisting of the training documents and the set of impostors.

As baselines, we use the top-ranked submissions and the meta-models combining all submissions of PAN-2014 and PAN-2015 shared tasks, as well as other recent studies which report AUC results in the same datasets. More specifically:

- Khonji & Iraqi (2014) [16]: This is a modification of GI [30] and the winning submission of PAN-2014 [34].
- Fréry et al. (2014) [9]: The second-best submission in PAN-2014, it is an eager verification approach using a decision tree classifier.
- META-PAN14 [34]: This is a simple heterogeneous ensemble reported by PAN organizers. It is based on the average of all 13 PAN-2014 submissions.
- Bagnall (2015) [2]: The winning approach of PAN-2015 [33]. It uses a multi-headed recurrent neural network language model.
- Moreau et al. (2015) [23]: The second-best submission of PAN-2015. It is an heterogeneous ensemble of 5 verification models.
- META-PAN15 [33]: A simple heterogeneous ensemble based on the average of all 18 PAN-2015 submissions.
- Potha & Stamatatos (2017) [26]: This is another modification of GI with improved results.
- Potha & Stamatatos (2018) [27]: This is a profile-based and intrinsic method using topic modeling.
- Ding et al. (2019) [7]: A neural network approach that jointly learns distributed word representations together with topical and lexical biases achieving improved results in PAN-2014 datasets.

4.3 Results

First, we compare the baselines with the stacking ensemble (based on a SVM meta-learner with its hyper-parameters tuned based on the training dataset[1]) as

[1] This is done for each PAN dataset separately. In all cases, an RBF kernel is selected.

Table 4. Evaluation results (AUC) on PAN-2014 datasets.

	DE	DR	EE	EN	GR	SP	Avg
Baselines							
Khonji & Iraqi (2014)	0.913	0.736	0.590	0.750	0.889	0.898	0.797
Fréry et al (2014)	0.906	0.601	0.723	0.612	0.679	0.774	0.716
META-PAN14 (2014)	0.957	0.737	0.781	0.732	0.836	0.898	0.824
Potha & Stamatatos (2017)	0.976	0.685	0.762	0.767	0.929	0.878	0.833
Potha & Stamatatos (2018)	0.982	0.646	0.781	0.761	0.919	0.902	0.832
Ding et al. (2019)	**0.998**	0.658	0.887	0.767	0.924	0.934	0.876
Proposed ensembles							
Stacking_{tuned}	0.988	**0.890**	0.861	0.832	0.969	0.922	0.910
Stacking_{fixed}	0.986	0.854	0.818	0.828	0.965	0.940	0.898
DES_{tuned}	0.983	0.879	0.876	0.843	0.973	0.945	0.916
DES_{fixed}	0.985	0.885	**0.901**	**0.857**	**0.977**	**0.963**	**0.928**

well as the proposed DES method. For the latter, thresholds are set to $a = 0.75$ and $b = 0.6$. Tables 4 and 5 present the evaluation results (AUC) per test dataset and the average performance over PAN-2014 and PAN-2015 datasets, respectively. Moreover, two versions of stacking and DES are reported: one using base verification models that have been tuned using the training dataset and another one using base models with fixed parameter settings.

As can be seen, the proposed DES method is the most effective one in most of the cases improving the best reported results for the specific datasets. Its performance is higher when fixed parameter settings are used in comparison to tuned models. This sounds reasonable since fixed models are less biased in the training dataset and the weight of each model is more reliably estimated. Nevertheless, DES based on tuned models also provides very good results. The stacking methods are also very effective surpassing in terms of average performance all baselines. In this case, the tuned models seem to be the best option. Again, this can be explained since the meta-learner needs as accurate base models as possible and tuned models are more likely to be more accurate than models with fixed settings. The improvement in average performance of the best ensemble models with respect to that of the best baselines is higher than 5% in the PAN-2014 datasets and more than 10% in the PAN-2015 datasets. It is also remarkable that the biggest improvement is achieved in datasets with very limited D_{known} size (PAN-2014-DR, PAN-2014-EE, PAN-2014-EN). All these indicate that the ensemble approach is much more reliable and effective in difficult verification cases where there are few known documents or documents belong to different domains.

We also examine the statistical significance of pairwise differences of all tested (both the proposed and baseline) methods using an approximate randomization test [24]. The null hypothesis assumes there is no difference between a pair of

Table 5. Evaluation results (AUC) on PAN-2015 datasets.

	DU	EN	GR	SP	Avg
Baselines					
Bagnall (2015)	0.700	0.811	0.882	0.886	0.820
Moreau et al. (2015)	0.825	0.709	0.887	0.853	0.819
META-PAN15 (2015)	0.696	0.786	0.779	0.894	0.754
Potha & Stamatatos (2017)	0.709	0.798	0.844	0.851	0.801
Potha & Stamatatos (2018)	0.572	0.764	0.859	0.946	0.785
Proposed Ensembles					
Stacking_{tuned}	0.858	0.864	0.955	0.976	0.913
Stacking_{fixed}	0.814	0.867	0.968	0.977	0.907
DES_{tuned}	0.849	**0.898**	0.962	0.971	0.920
DES_{fixed}	**0.866**	0.879	**0.988**	**0.990**	**0.930**

tested methods when each PAN dataset is considered separately. The baseline approach of Ding et al. (2019) [7] is not included in these tests since we did not have access to the original output of this method for each individual verification instance. In most of the cases, the proposed ensembles are significantly ($p < 0.05$) better than the baselines. Notable exceptions are PAN-2014-DE, where the proposed ensembles are not significantly better than the baselines of Potha and Stamatatos (2017) and Potha and Stamatatos (2018), and PAN-2015-DU, where the difference with Moreau et al. (2015) is not significant. On the other hand, the differences between the stacking and DES ensembles in most of the cases are not statistically significant. The full results of this analysis are not included here due to lack of space.

Next, we focus on the effect of thresholds a, b in the average performance of DES. Figure 1 depicts the average AUC (for all PAN-2014 and PAN-2015 datasets) of DES using either tuned or fixed base models for a range of of threshold a values while we fix threshold $b = 0.6$. Recall that the higher threshold a is, the less similar training instances are retrieved. In case of very high values of a it is possible that the retrieved set of training instances is empty. In such cases, the test instance is left unanswered by getting a fix verification score=0.5. This is in accordance with the evaluation setup of PAN shared tasks [33, 34]. From the obtained results, it is clear that the fixed models are better than the tuned models in almost all examined cases. In addition, DES is clearly better than the best baseline for the whole range of threshold a values. With respect to the best stacking ensemble, it seems that DES is better in both datasets when $0.7 \leq a \leq 0.9$. This means that a should be set to a relatively large value to filter out most dissimilar training instances.

Figure 2 depicts the corresponding average performance of DES method, based either on tuned or fixed models, on PAN-2014 and PAN-2015 datasets varying threshold b while we fix threshold $a = 0.75$. Similar to the previous case, for high values of b, if none of the verifiers is selected for a test instance, then

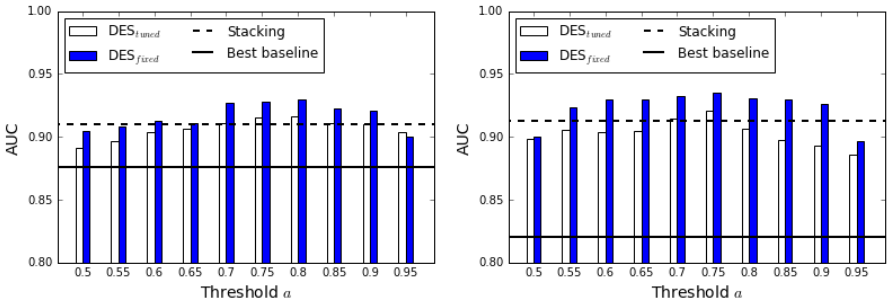

Fig. 1. Average AUC of DES method, using either tuned or fixed base models, on PAN-2014 (left) and PAN-2015 (right) datasets for varying threshold a. The performances of the best stacking ensemble (based on tuned models) and the best baseline are also shown.

it is assigned a fix verification score=0.5, namely it is left unanswered. Again, DES based on tuned models is outperformed by the DES using fixed models in almost all b values. In this case, this difference is higher in comparison to that of Fig. 1. This clearly shows that the tuned models are more biased in the training dataset and, therefore, less useful in DES. It is also important that the performance of DES remains better than the best baseline for the whole range of examined b values. As concerns the comparison to the stacking ensemble, we see that DES is better when $b \leq 0.6$. It seems that the performance of DES remains robust when b decreases, even in case it is set to zero. In that extreme case, all base models are taken into account. However, some of them (the ones with poor performance in the selected training instances) will be considered with very small weight, so practically they are filtered out. Actually, when $b = 0$ DES achieves comparatively good results. This means that it is possible to reduce the parameters of DES by setting $b = 0$ (use all base models) and still getting respectable performance.

5 Discussion

In this paper, we present author verification approaches based on ensemble learning. We collect a relatively large pool of 47 base verifiers covering the basic paradigms in this area, namely, both intrinsic and extrinsic methods as well as both instance-based and profile-based methods. This collection of verifiers provides a pluralism of verification scores and we attempt to take advantage of their correlations by building two ensembles. The first one is based on stacked generalization and learns patterns of agreement/disagreement among verifiers. In other words, it learns when to trust a verifier. The second, more sophisticated approach, is based on dynamic ensemble selection and attempts to find relevant training instances with respect to each verification case separately and

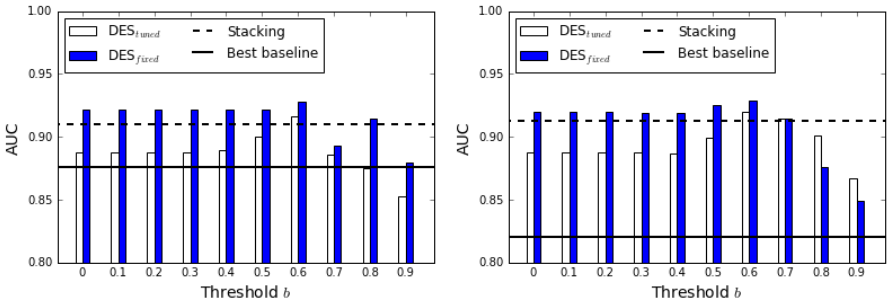

Fig. 2. Average AUC of DES method, using either tuned or fixed base models, on PAN-2014 (left) and PAN-2015 (right) datasets for varying threshold b. The performances of the best stacking ensemble (based on tuned models) and the best baseline are also shown.

then filters out verifiers that are not too specialized for the selected subset of instances.

Both ensemble approaches outperform a set of strong baselines according to experiments using ten PAN benchmark datasets. The performance of DES is (in average) more than 5% better than the best baseline in PAN-2014 datasets and more than 10% better in PAN-2015 datasets. Recall that PAN-2015 datasets consist of difficult cross-domain cases (where the known and unknown documents are about distant topics or belong to different genres). In addition, the performance of the proposed ensembles is much better than the strongest baseline in datasets where only one known document is provided (PAN-2014-DR, PAN-2014-EN, PAN-2015-EN). This indicates that our ensembles are able to handle challenging verification scenarios and are more robust than individual models.

DES has two parameters that control how many similar training instances will be retrieved and how many base classifiers will be considered. It has been shown that a relatively high threshold a value is required to filter out most irrelevant training instances. In addition, relatively good results are obtained when b takes low values including the case where $b = 0$. This means it does not harm to consider all possible base models given that their weight (determined by their performance on the similar training instances) will be quite low.

Our experiments demonstrate that the stacking ensemble works better with base models that are tuned to maximize performance in the training dataset. On the other hand, the DES method is more effective when fixed parameter settings are used in the base models. Tuned verifiers are biased in the training dataset and the estimation of their weight within DES becomes less reliable. This could be used to further enrich the pool of our base verifiers considering several versions of the same approach with different fixed parameter settings. Another possible future work direction is to try to combine the stacking and DES ensembles in a more complex approach.

References

1. Almishari, M., Oguz, E., Tsudik, G.: Fighting authorship linkability with crowdsourcing. In: Proceedings of the second ACM conference on Online social networks, COSN. pp. 69–82 (2014)
2. Bagnall, D.: Author identification using multi-headed recurrent neural networks. In: Cappellato, L., Ferro, N., Gareth, J., San Juan, E. (eds.) Working Notes Papers of the CLEF 2015 Evaluation Labs (2015)
3. Barbon, S., Igawa, R., Bogaz Zarpelão, B.: Authorship verification applied to detection of compromised accounts on online social networks: A continuous approach. Multimedia Tools and Applications 76(3), 3213–3233 (2017)
4. Bartoli, A., Dagri, A., Lorenzo, A.D., Medvet, E., Tarlao, F.: An Author Verification Approach Based on Differential Features. In: Cappellato, L., Ferro, N., Gareth, J., San Juan, E. (eds.) Working Notes Papers of the CLEF 2015 Evaluation Labs (2015)
5. Brocardo, M., Traore, I., Woungang, I., Obaidat, M.: Authorship verification using deep belief network systems. International Journal of Communication Systems 30(12) (2017)
6. Castro-Castro, D., Arcia, Y.A., Brioso, M.P., Guillena, R.M.: Authorship verification, average similarity analysis. In: Recent Advances in Natural Language Processing. pp. 84–90 (2015)
7. Ding, S., Fung, B., Iqbal, F., Cheung, W.: Learning stylometric representations for authorship analysis. IEEE Transactions on Cybernetics 49(1), 107–121 (2019)
8. Duman, S., Kalkan-Cakmakci, K., Egele, M., Robertson, W., Kirda, E.: Emailprofiler: Spearphishing filtering with header and stylometric features of emails. In: Proceedings - International Computer Software and Applications Conference. vol. 1, pp. 408–416 (2016)
9. Fréry, J., Largeron, C., Juganaru-Mathieu, M.: UJM at CLEF in author identification. Proceedings CLEF-2014, Working Notes pp. 1042–1048 (2014)
10. Halvani, O., Graner, L., Vogel, I.: Authorship verification in the absence of explicit features and thresholds. In: European Conference on Information Retrieval. pp. 454–465. Springer (2018)
11. Hernández, C.Á., Calvo, H.: Author verification using a semantic space model. Computación y Sistemas 21(2) (2017)
12. Hürlimann, M., Weck, B., van den Berg, E., Šuster, S., Nissim, M.: GLAD: Groningen Lightweight Authorship Detection. In: Cappellato, L., Ferro, N., Jones, G., San Juan, E. (eds.) CLEF 2015 Evaluation Labs and Workshop – Working Notes Papers. CEUR-WS.org (2015)
13. Jankowska, M., Milios, E., Keselj, V.: Author verification using common n-gram profiles of text documents. In: Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers. pp. 387–397 (2014)
14. Juola, P., Stamatatos, E.: Overview of the author identification task at PAN 2013. In: Working Notes for CLEF 2013 Conference (2013)
15. Kestemont, M., Luyckx, K., Daelemans, W., T., C.: Cross-genre authorship verification using unmasking. English Studies 93(3), 340–356 (2012)
16. Khonji, M., Iraqi, Y.: A slightly-modified GI-based author-verifier with lots of features (asgalf). In: CLEF 2014 Labs and Workshops, Notebook Papers. CLEF and CEUR-WS.org (2014)

17. Ko, A.H., Sabourin, R., de Souza Britto Jr., A.: From dynamic classifier selection to dynamic ensemble selection. Pattern Recognition 41(5), 1718–1731 (2008)
18. Kocher, M., Savoy, J.: A simple and efficient algorithm for authorship verification. Journal of the Association for Information Science and Technology 68(1), 259–269 (2017)
19. Koppel, M., Schler, J., Argamon, S., Winter, Y.: The fundamental problem of authorship attribution. English Studies 93(3), 284–291 (2012)
20. Koppel, M., Schler, J., Bonchek-Dokow, E.: Measuring differentiability: Unmasking pseudonymous authors. Journal of Machine Learning Research 8, 1261–1276 (2007)
21. Koppel, M., Winter, Y.: Determining if two documents are written by the same author. Journal of the American Society for Information Science and Technology 65(1), 178–187 (2014)
22. Layton, R., Watters, P., Ureche, O.: Identifying faked hotel reviews using authorship analysis. In: Proceedings - 4th Cybercrime and Trustworthy Computing Workshop, CTC 2013. pp. 1–6 (2013)
23. Moreau, E., Jayapal, A., Lynch, G., Vogel, C.: Author verification: Basic stacked generalization applied to predictions from a set of heterogeneous learners-notebook for pan at clef 2015. In: CLEF 2015-Conference and Labs of the Evaluation forum. CEUR (2015)
24. Noreen, E.: Computer-Intensive Methods for Testing Hypotheses: An Introduction. A Wiley-Interscience publication, Wiley (1989)
25. Potha, N., Stamatatos, E.: A profile-based method for authorship verification. In: Artificial Intelligence: Methods and Applications - Proceedings of the 8th Hellenic Conference on AI, SETN. pp. 313–326 (2014)
26. Potha, N., Stamatatos, E.: An improved impostors method for authorship verification. Proc. of the 8th International Conference of the CLEF Association, CLEF 2017, Lecture Notes in Computer Science 10456, 138–144 (2017)
27. Potha, N., Stamatatos, E.: Intrinsic author verification using topic modeling. In: Artificial Intelligence: Methods and Applications - Proceedings of the 10th Hellenic Conference on AI, SETN (2018)
28. Potha, N., Stamatatos, E.: Improving author verification based on topic modeling. Journal of the Association for Information Science and Technology (2019)
29. Sanderson, C., Guenter, S.: Short text authorship attribution via sequence kernels, markov chains and author unmasking: An investigation. In: Proceedings of the International Conference on Empirical Methods in Natural Language Engineering. pp. 482–491 (2006)
30. Seidman, S.: Authorship Verification Using the Impostors Method. In: Forner, P., Navigli, R., Tufis, D. (eds.) CLEF 2013 Evaluation Labs and Workshop – Working Notes Papers (2013)
31. Stamatatos, E.: A Survey of Modern Authorship Attribution Methods. Journal of the American Society for Information Science and Technology 60, 538–556 (2009)
32. Stamatatos, E.: Authorship verification: A review of recent advances. Research in Computing Science 123, 9–25 (2016)
33. Stamatatos, E., Daelemans, W., Verhoeven, B., Juola, P., López-López, A., Potthast, M., Stein, B.: Overview of the author identification task at PAN 2015. In: Working Notes of CLEF 2015 - Conference and Labs of the Evaluation forum (2015)
34. Stamatatos, E., Daelemans, W., Verhoeven, B., Potthast, M., Stein, B., Juola, P., Sanchez-Perez, M., Barrón-Cedeño, A.: Overview of the author identification task at pan 2014. In: CLEF Working Notes. pp. 877–897 (2014)

35. Stover, J.A., Winter, Y., Koppel, M., Kestemont, M.: Computational authorship verification method attributes a new work to a major 2nd century african author. Journal of the American Society for Information Science and Technology 67(1), 239–242 (2016)
36. Tuccinardi, E.: An application of a profile-based method for authorship verification: Investigating the authenticity of Pliny the Younger's letter to Trajan concerning the Christians. Digital Scholarship in the Humanities 32(2), 435–447 (2017)
37. Wolpert, D.H.: Stacked generalization. Neural Networks 5, 241 – 259 (1992)