An Author Verification Approach Based on Differential Features

Notebook for PAN at CLEF 2015

Alberto Bartoli, Alex Dagri, Andrea De Lorenzo,
Eric Medvet, and Fabiano Tarlao

DIA - University of Trieste, Italy
bartoli.alberto@univ.trieste.it, postaxmi@gmail.com, andrea.delorenzo@units.it,
emedvet@units.it, fabiano.tarlao@phd.units.it

Abstract We describe the approach that we submitted to the 2015 PAN competition [7] for the author identification task[1]. The task consists in determining if an unknown document was authored by the same author of a set of documents with the same author.
We propose a machine learning approach based on a number of different features that characterize documents from widely different points of view. We construct non-overlapping groups of homogeneous features, use a random forest regressor for each features group, and combine the output of all regressors by their arithmetic mean. We train a different regressor for each language.
Our approach achieved the first position in the final rank for the Spanish language.

1 Problem statement

A problem instance is a tuple $\langle K, u, L\rangle$ where K is a set of documents $\{k_1, \ldots, k_n\}$ authored by the same author (called known documents), u is a document whose authorship must be ascertained (called unknown document), L is an enumerated value specifying the language of the documents: English, Dutch, Greek or Spanish. All documents in a problem instance are in the same language.

The author verification procedure consists in generating an answer in the form of a real number in $[0, 1]$ which quantifies the degree of confidence of being u authored by the same author of the documents in K: 0 indicates absolute certainty that u was not authored by the same author of documents in K, while 1 indicates absolute certainty that all documents were authored by the same author.

A set of solved problem instances (the *training set*) is available in which, for each problem instance $\langle K, u, L\rangle$, the solution consisting in one between 0 and 1 is provided.

[1] During the competition we discovered several opportunities for fraudulently boosting the accuracy of our method during the evaluation phase. We will describe these opportunities in a future report. We notified the organizers which promptly acknowledged the high relevance of our concerns and took measures to mitigate the corresponding vulnerabilities. The organizers acknowledged our contribution publicly. We submitted for evaluation an honestly developed method—the one described in this document—that did not exploit such unethical procedures in any way.

The effectiveness of a method for author identification is assessed using a *testing set* of solved problem instances, as follows. The answers generated by the method for the problem instances in the testing set are compared against the actual values and the comparison outcome is expressed in terms of two indexes: area under the ROC curve (AUC) and $c@1$. AUC is computed basing on the ROC curve plotted by comparing the generated answers against a threshold moving between 0 and 1, hence obtaining a binary classification task. The latter index is computed as $c@1 = \frac{n_c}{n} + \frac{n_u n_c}{n^2}$, where n is the size of the testing set, n_u is the number of unanswered problem instances (i.e., those for which the generated answer was exactly 0.5), n_c is the number of correct answers (i.e., those for which the generated answer > 0.5 and the actual answer is 1 and those for which the generated answer < 0.5 and the actual answer is 0).

2 Our approach

We propose a machine learning approach based on a number of different features that characterize documents from widely different points of view: character, word, part-of-speech, sentence length, punctuation. We construct non-overlapping groups of homogeneous features and use a random forest regressor for each features group. The output of the resulting ensemble of regressors is the arithmetic mean of the output generated by each random forest.

We train a different regressor for each language. Based on extensive experimention on the training set, we decided to use the same features for problem instances in Dutch, Greek, Spanish but a different set of features for problem instances in English.

2.1 Features

We extract a number of different features from each document. For ease of presentation, we group homogeneous features together, as described below.

Word $n$grams (WG) We convert all characters to lowercase and then we transform the document to a sequence of words. We consider white spaces, punctuation characters and digits as word separators. We count all word $n$grams, with $n \leq 3$, and we obtain a feature for each different word $n$gram which occurs in the training set documents of a given language.

Character $n$grams (CG) We replace punctuation characters and digits with blank spaces and then sequences of blank spaces with a single blank space. We count all character $n$grams, with $n \leq 3$, and we obtain a feature for each different character $n$gram which occurs in the training set documents of a given language.

POS (part-of-speech) tag $n$grams (PG) We apply a *part of speech (POS) tagger* on each document, which assigns words with similar syntactic properties to the same POS tag. For English and Dutch we use the Apache OpenNLP Tools[2], for Greek we use the tagger developed by the Department of Informatics at Athens University

[2] `http://opennlp.apache.org`

of Economics and Business[3] while for Spanish we use TreeTagger[4] [6]. We count all POS $n$grams, with $n \leq 3$, and we obtain a feature for each different POS $n$gram which occurs in the training set documents of a given language.

Word lengths (WL) We convert all characters to lowercase and then we transform the document to a sequence of words. We consider white spaces, punctuation characters and digits as word separators. We count the number of words whose length in characters is n, with $n \in \{1, \ldots, 16\}$: we obtain a feature for each value of n.

Sentence lengths (SL) We transform the document to a sequence of tokens, a token being a sequence of characters separated by one or more blank spaces. Next, we transform the sequence of tokens to a sequence of sentences, a sentence being a sequence of tokens separated by any of the following characters: `.`,`;`,`:`,`!`,`?`. We count the number of sentences whose length in tokens is n, with $n \in \{1, \ldots, 40\}$: we obtain a feature for each value of n.

Sentence length $n$grams (SG) We transform each document to a sequence of labels, where each label represents a full sentence and is chosen based on the sentence length (as described in the following). Next, we compute the $n$grams of the resulting labels, with $n \leq 2$. In detail, we execute a preliminary analysis of all documents of a given language in the training set, as follows. For each document, we transform the document to a sequence of sentences as illustrated for the SL features group. Next, we compute the distribution of sentence length across all sentences in the training set and determine the length values associated with the following percentile values: 10%, 25%, 75%, and 90%. In other words, we divide the range of sentence lengths observed in the training set in 5 intervals, with boundaries between intervals determined by the specified percentiles. The label we assign to each sentence corresponds to one of the 5 lenght intervals, i.e., $]0\%, 10\%]$, $]10\%, 25\%]$, and so on: we obtain a feature for each label $n$grams which occurs in the training set documents of a given language.

Word richness (WR) We transform the document to a sequence of words as for the WG features group. Then we compute the ratio between the number of distinct words and the number of total words in the document—this features group contains only one feature.

Punctuation $n$grams (MG) We transform the document by removing all characters not included in the following set: {`,`, `.`, `;`, `:`, `!`, `?`, `"`}—the resulting document thus consists of a (possibly empty) sequence of characters in that set. We then count all character $n$grams of the resulting document, with $n \leq 3$, and we obtain a feature for each different punctuation $n$gram which occurs in the training set documents of a given language.

Text shape $n$grams (TG) We transform the document as follows: sequences of digits are replaced by the single character `n`; sequences of alphabetic characters are replaced by a single character: `l` if all the characters in the sequence are lowercase, `u` if only the first character is uppercase, `w` if at least two characters are uppercase; sequences of blank spaces are replaced by a single blank space; other characters are left unchanged. We then count all character $n$grams of the resulting document, with

[3] `http://nlp.cs.aueb.gr/software.html`

[4] `http://www.cis.uni-muenchen.de/~schmid/tools/TreeTagger`

$n \leq 3$, and we obtain a feature for each different character $n$gram which occurs in the training set documents of a given language.

2.2 Feature selection, normalization, and aggregation

We perform a simple *feature selection* for features in groups WG, CG, PG, and TG. To this end, we apply the following procedure to each of the 4 partitions of the training set for which the language of the documents was the same—in other words, we select different features for each language. We compute the feature values for all the documents in the training set partition. Next, among each group, we sort the features according to their average values on the documents of the partition—greater values coming first. Finally, for each group, we keep the n_{sel} top features. We set $n_{\text{sel}} = 500$ for WG, CG and PG and $n_{\text{sel}} = 100$ for TG.

After the feature selection, we perform a normalization of the features values, as follows. Let $f_i(d)$ be the value of the ith feature for the document d and let G be the group of features (as defined in Section 2.1) which includes the feature f_i, we set $f_i(d) := \frac{f_i(d)}{\sum_{f_j \in G} f_j(d)}$. We execute this procedure for all the groups of features, except for WR, which consists of a single feature.

Finally, for the purpose of obtaining a single feature vector for each problem instance, rather than one feature vector for each document, we build a new feature f'_i whose value is obtained from the values of the corresponding f_i for the documents in K and the document u, as follows:

$$f'_i(\langle K, u, L\rangle) = \text{abs}\left(f_i(u) - \frac{\sum_{k \in K} f_i(k)}{|K|}\right) \tag{1}$$

In other words, we consider the absolute difference between the feature value for the unknown document u and the average of the feature values for the known documents in K. We also consider a variant of our approach in which the difference is divided by the feature value for u:

$$f''_i(\langle K, u, L\rangle) = \frac{f'_i(\langle K, u, L\rangle)}{f_i(u)} \tag{2}$$

2.3 Regressor

We explored three different regressor algorithms: trees (Tree), random forests (RF), and support vector machines (SVM). In particular, we use the algorithm proposed in [5] for Tree, we use the gaussian kernel and $C = 1$ for SVM [4], and we use the algorithm for regression proposed in [3] with $n_{\text{tree}} = 500$ for RF.

We apply each regressor, both in training and actual regression phase, only to the feature values of the same group. For obtaining an answer in $[0, 1]$ for a problem instance, we average the predictions obtained by the trained regressors on the features groups. In other words, we built an *ensemble* of *group regressors*.

3 Analysis

As described in the previous section, we considered two set of features (f' and f'') and 3 regressors. We systematically assessed the effectiveness of all the 6 resulting combinations by means of a *leave-one-out* procedure applied on the training set, separately for each language. That is, for each language, type of feature, and regressor, (i) we built the subset T of the problem instances of the training with that language, (ii) we removed one element t_0 from T, (iii) we computed the feature values for the problem instances in T and trained the regressor, (iv) we applied the trained regressor to the problem instance t_0 and compared the generated answer against the known one. We repeated all but first steps $|T| = 100$ times, i.e., by removing each time a different element, and computed the performance of the method in terms of the indexes defined in Section 1: $c@1$ and AUC.

The results are in Table 1: the table shows $c@1$ and AUC values for each method, the method name being composed by the regressor acronym and one among abs or rel indicating the use of f' or f'' features, respectively. It can be seen that RF provides

	$c@1$				AUC			
Method	EN	DU	GR	SP	EN	DU	GR	SP
RF-abs	0.67	0.74	0.77	0.94	0.718	0.707	0.808	0.992
RF-rel	0.58	0.66	0.77	0.95	0.584	0.776	0.796	0.989
SVM-abs	0.48	0.67	0.69	0.92	0.513	0.707	0.754	0.978
SVM-rel	0.45	0.62	0.66	0.86	0.584	0.645	0.732	0.936
Tree-abs	0.69	0.70	0.53	0.94	0.725	0.708	0.557	0.951
Tree-rel	0.56	0.62	0.69	0.97	0.526	0.595	0.699	0.992

Table 1. $c@1$ and AUC for 6 methods.

in general better results than the other regressors; moreover, RF-abs appears to be the best performing method. In order to further validate the latter finding, we performed a Wilcoxon signed-rank test [1] with a significance level of 5% and Bonferroni correction [2]: the outcome is that RF-abs is significantly better than all the other methods, except Tree-rel, for a little gap, and RF-rel; RF-rel is not better than the other methods except SVM-rel; Tree-rel is not better than all the other methods.

In order to gain insights about which features group appeared to be more suitable for accomplishing the considered task, we applied the RF-abs method (with the leave-one-out procedure described above) 9 times, each time removing one of the 9 features groups—i.e., we performed a features group ablation analysis. The results (in terms of $c@1$) are reported in Table 2. It can be seen, by comparing results of method RF-abs with those of Table 1, that the largest decrease of $c@1$ occurs by removing features group MG, while the smallest one occurs by removing features group WR—on the average around 3% and 1%, respectively. It can also be observed that feature ablation may actually lead to some improvements: for English, we obtain 0.69, rather than 0.67, by removing WG; for Spanish, we obtain 0.96, rather than 0.94, by removing either WG or WR.

Features groups	EN	DU	GR	SP
All-WG	0.69	0.68	0.75	0.96
All-CG	0.66	0.71	0.75	0.95
All-PG	0.68	0.70	0.75	0.94
All-WL	0.67	0.71	0.75	0.95
All-SL	0.65	0.70	0.73	0.95
All-SG	0.66	0.69	0.75	0.95
All-WR	0.67	0.71	0.75	0.96
All-MG	0.62	0.71	0.74	0.94
All-TG	0.63	0.72	0.75	0.93

Table 2. $c@1$ with RF-abs by removing one features group at once.

Then, we analyzed the performance of RF-abs in terms of feature addition. We considered RF-abs using only features group MG (which showed to be the most relevant, according to the ablation analysis) and then using only MG and each of the 8 other features groups in isolation. The results are reported in Table 3. It can be seen

Features groups	EN	DU	GR	SP
MG	0.71	0.63	0.66	0.89
MG+WG	0.67	0.71	0.75	0.94
MG+CG	0.73	0.63	0.68	0.93
MG+PG	0.71	0.67	0.68	0.94
MG+WL	0.72	0.65	0.66	0.91
MG+SL	0.73	0.65	0.72	0.90
MG+SG	0.73	0.58	0.71	0.87
MG+WR	0.59	0.56	0.60	0.74
MG+TG	0.72	0.64	0.68	0.91

Table 3. $c@1$ with RF-abs by using MG features and zero or one other features group.

that, for English, there are combinations that improve $c@1$ with respect to the baseline value 0.67: MG+CG, MG+SL, and MG+SG reach 0.73. Since such improvement is not negligible, we inspected the mutual effect of these features groups more closely by analyzing the $c@1$ values resulting from all their combinations. The results are: 0.78 with MG+CG+SL, 0.65 with CG+SG+SL, 0.71 with MG+SL+SG, and 0.73 with MG+CG+SL+SG. Based on these results, which improved the 0.67 baseline (all feature groups), we chose to use RF-abs with only 3 features groups (MG+CG+SL), only for the English language. On the other hand, we did not notice significant improvements for specific sets of features groups for the other languages: hence, for Dutch, Greek, and Spanish, we chose to use RF-abs with all the features groups.

We observed that the results for the Spanish language tend to be much better than for the other languages. We believe that such good results depend more on the peculiarity of this dataset rather than to the quality of our method: indeed the training set for Spanish

contained 100 problem instances with 5 documents each, but the number of distinct documents, though, was only 42.

3.1 Final results

Table 4 reports the final results obtained in the competition, as released by the organizers[5]. The table shows the performance indexes computed on a separated testing set which was not available during the method design phase. Besides $c@1$ and AUC, the table also reports a score, according to which a ranking for each language has been compiled: the score is the product of $c@1$ and AUC.

Method	Language	$c@1$	AUC	Score	Ranking
RF-abs on MG+CG+SL	EN	0.56	0.578	0.323	10/18
RF-abs on all	DU	0.69	0.751	0.518	4/17
RF-abs on all	GR	0.66	0.698	0.459	7/14
RF-abs on all	SP	0.83	0.932	0.773	1/17

Table 4. Final results.

References

1. Bauer, D.F.: Constructing confidence sets using rank statistics. Journal of the American Statistical Association 67(339), 687–690 (1972)
2. Benjamini, Y., Hochberg, Y.: Controlling the false discovery rate: a practical and powerful approach to multiple testing. Journal of the Royal Statistical Society. Series B (Methodological) pp. 289–300 (1995)
3. Breiman, L.: Random forests. Machine learning 45(1), 5–32 (2001)
4. Chang, C.C., Lin, C.J.: Libsvm: a library for support vector machines. ACM Transactions on Intelligent Systems and Technology (TIST) 2(3), 27 (2011)
5. Loh, W.Y.: Classification and regression trees. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery 1(1), 14–23 (2011)
6. Schmid, H.: Probabilistic part-of-speech tagging using decision trees. In: Proceedings of the international conference on new methods in language processing. vol. 12, pp. 44–49. Citeseer (1994)
7. Stamatatos, E., Daelemans, W., Verhoeven, B., Juola, P., Lopez Lopez, A., Potthast, M., Stein, B.: Overview of the Author Identification Task at PAN 2015. In: Working Notes Papers of the CLEF 2015 Evaluation Labs. CEUR Workshop Proceedings, CLEF and CEUR-WS.org (Sep 2015), http://www.clef-initiative.eu/publication/working-notes

[5] http://www.tira.io/task/authorship-verification/