Cross-Domain Authorship Attribution Combining Instance-Based and Profile-Based Features

Notebook for PAN at CLEF 2019

Andrea Bacciu, Massimo La Morgia, Alessandro Mei, Eugenio Nerio Nemmi, Valerio Neri, and Julinda Stefa

Affiliation
Department of Computer Science, Sapienza University of Rome, Italy
{lamorgia, mei, nemmi, stefa}@di.uniroma1.it
bacciu.1747105@studenti.uniroma1.it, neri.1754516@studenti.uniroma1.it

Abstract Being able to identify the author of an unknown text is crucial. Although it is a well-studied field, it is still an open problem, since a standard approach has yet to be found. In this notebook, we propose our model for the Authorship Attribution task of PAN 2019, that focuses on cross-domain setting covering 4 different languages: French, Italian, English, and Spanish. We use n-grams of characters, words, stemmed words, and distorted text. Our model has an SVM for each feature and an ensemble architecture. Our final results outperform the baseline given by PAN in almost every problem. With this model, we reach the second place in the task with an F1-score of 68%.

1 Introduction

Stylometry is the application of the study of linguistic style, and it is often used for establishing authenticity or authorship of an unknown text. It was first used to determine the author of unknown playwrights, but it soon became a powerful tool in Forensics analysis. Nowadays, it is also used in court. It is famous how linguistic analysis helped, for example, to solve the Coleman case [12]. PAN 2019 [4,9] focuses on the task of Authorship Attribution, that is the task of determining the author of a disputed document given a set of known authors. When the disputed text could be written by an author that does not belong to the known set, the task is called open-set Authorship Attribution. This task is more challenging than the closed-set, where we always know that some author in our set wrote the disputed document. The main difference is that in the closed-set we look for the most similar author, while in the open-set, we must further understand if the most similar author is also the real author. Since for each author we have more training documents, we build features using them individually and then concatenating them. We refer to the features computed on a single document as instance-based features while to those concatenated as profile-based features. We choose to combine these two methods in order to catch both the patterns that they capture. In this paper, we extract different types of features and we pre-process the texts in several ways. We use a Tokenizer, a Stemmer, and a POS tagger. Afterward, starting from the pre-processed text, we extract 6 types of different features, and we use the Tf-Idf to weight them. We train an SVM for each feature, then we ensemble their predictions with the soft-voting. Our architecture uses 4 SVMs for the French, 5 for the Spanish and the Italian, and 6 for the English. With our technique, we achieve a final overall F1-score of 70.5%.

2 Related Work

Typically, we can divide the approaches to solve the Authorship Attribution problem in two different categories: profile-based or instance-based approach [19]. In the profile-based approach, we concatenate together texts of the same author to extract its stylometric profile. This method relies on collecting as more information of the user as possible, in one document. It is especially suited when the texts are similar in the domain space. In the instance-based approach, instead, we analyze the texts associated with an author separately. It gives better results if the texts associated with each author are from different domains. In both these approaches, stylometric features are prevalent.

Stylometry is one of the most effective technique to distinguish the authorship of a text, and it has been used in most of the studies. Although to extract a stylometric profile there are several kinds of features, some of them have been proven to be more powerful. Several independent Authorship Attribution studies showed that character n-gram are key features and they are more robust compared to the word n-gram features in cross-topic and cross-genre conditions [13,19,20]. This is because character n-grams can capture the use of punctuation marks, the lexical information, the use of the capital letter, and it is also tolerant of typing errors.

Despite the effectiveness of character n-gram applied on plain text, several variations have been proposed to improve it. One of the most successful variations consists in text distortion (e.g., by removing some character) [21]. There are a variety of features that can represent the stylistic profile of an author, such as Function Words, POS tagging, and word n-gram. Some studies proposed to combine these features to obtain a stronger representation of the author stylistic profile. Custodio et al. [3] proposed the most interesting approach. It consists in fitting three different SVMs with different features then combine their predictions. While the features mentioned above work differently based on configuration and language, some method does not. Compression [5, 6, 14, 19] is a particularly attractive method because it does not need any configuration, it is language independent, easy to apply, and does not require any prior knowledge of features extraction to produce a prediction. The drawback is the high cost in terms of computational expensiveness. The compression method focuses on repetitions and the ability of the compression algorithm to detect them. It uses off-the-shelf compression algorithms to compress all the texts of the same author together. The prediction is evaluated calculating the similarity between the compressed unknown text and the compression of each file of the author concatenated.

Koppel et al. [11], introduced another innovative approach called unmasking. It is a complex meta-learning approach to Author Verification, especially suited for long texts. The idea behind this method is that, most of the time, few features are the most significant. To show this, they use an SVM classifier to determine the accuracy results of the cross-validation between known versus unknown documents. Then, they compute the most significant features, remove them, and repeat the process. After some iterations, they show that if the predicted author is the same of the unknown document, the accuracy of the cross-validation decreased significantly, while if the author is not the same, the accuracy remains pretty high. Kestemont et al [8] used the unmasking method on two datasets, one intra-genre, and the other cross-genre. For the first dataset, they confirmed the effectiveness of the methodology, while for the cross-genre dataset, they show that the reliability of this approach drastically drops. Abbasi et al. [1] tried unsupervised methods to perform user Authorship Verification. They used a set of features comprising Lexical, Syntactic, Structural, Content and Idiosyncratic types over 4 different datasets with the best result reached on the eBay comments dataset with a 96% accuracy over 25 users. Kocher et al. [10] used an unsupervised method too. They use the 200 most common terms of the disputed text (isolated words and punctuation symbols) and simple distance measure and a set of impostors, to determine whether or not the disputed text was written by the proposed author. They use a dataset with texts of four different languages: English, Dutch, Spanish, and Greek. Their result was of 70% accuracy. Shrestha et al. [18] were the first to attempt to solve this task using a Convolutional Neural Network with character n-grams as input. They used three layers: A character embedding layer, a convolutional layer, and a fully connected softmax layer. With this architecture, they score an accuracy of 72% as best result, on a dataset of 50 users. Recently, Brocardo et al. [2] used Deep Belief Network to perform the task. They introduced new stylometric features and a method to merge pairs of random features. They combine two similar features by computing cosine similarity for every pair and then, they apply a linear combination to reduce two features into one. They use the Equal Error Rate (EER) as evaluation metric, reaching a value of 8.2% and 5.4% for the forgery dataset. Ruder et al. [17] perform a comparison between several Convolutional Neural Network approaches and the traditional ones. Their best model outperforms the results of three out of five datasets they tested on. Potha et al. [15] propose a variation of the Common N-Grams (CNG) approach originally proposed by Keselj et al. [7] for closed-set attribution and later modified by Stamatatos [21]. Following the profile-based paradigm, this method firstly concatenates all samples of known authorship into a single document and then extracts the representation vector using character n-gram. Another vector is produced from the disputed document and the two vectors are compared using a dissimilarity function. If the resulting score is above a certain threshold, the authorship of the disputed document is assigned to the author of the known documents. They tested this model on PAN 2013 dataset, with an overall F1 score of 0.78%.

3 Problem Background

The main idea behind the Authorship Attribution is that by extracting some stylistic or syntactic features, we can distinguish between texts written by different authors. The 2019 edition of PAN shared task, focuses on finding the author of a series of fanfictions belonging to different fandoms. Fanfiction is a work of fiction written by a fan of another work. The authors of the fanfictions usually maintain the original characters and the story setting or add their elements. Fandom is the domain to which the fanfiction belongs to. For example, if the fanfiction is based on the Harry Potter saga, this means it belongs to the fandom of Harry Potter. In the dataset, there are a set $\mathcal{D}$ of known fanfictions and a set $\mathcal{U}$ of unknown fanfictions. The goal is to identify for each document in $\mathcal{U}$ the correct author. The correct author can be one of the writers of the documents in $\mathcal{D}$ or none of them. The nature of the dataset rises three main problems. The authors are fandom writer, so they try to reproduce the writing style of the original work. The fandom of the known document and the fandom of the unknown documents are different. Finally, we are in an open-set problem, so there is no certainty that the unknown text is written by one of the authors in the known set.

4 Dataset

Before starting to address the problem, we collect some statistics about the PAN dataset, to better understand the data. PAN dataset is divided into 20 problems, 5 problems for each language: English, French, Italian, Spanish. For each problem, we have 9 known authors with 7 documents each, for a total of 63 training texts. The number of texts for which we have to predict an author is not the same for every problem. They vary with a maximum of 561 documents in the first problem and a minimum of 38 for the tenth problem. On average we have 202 of unknown fanfics. Since it is well known that in the Authorship Attribution problem, the number of words per document directly affect the performances of the classifiers, we analyze the distribution of the words in each document with known authors. We notice that inside the same problem the length of the documents can be very different. Globally, almost all the documents are in a range between 500 and 1000 words, with the shortest document of 382 words and the longest of 1523.

5 Authorship Attribution

In the subsections below, we firstly describe our steps to prepare the data and the tool we used. Later, we describe which features we choose to perform the classification, and finally, we describe our proposed model to solve the Authorship Attribution task.

5.1 Text Pre-Processing

Pre-processing is a crucial step to prepare the data in almost every NLP problems. Text pre-processing usually consists in normalize, sanitize or alter the text to remove noise, error, or completely change the data format. We pre-process the texts using different techniques, following we briefly describe the text pre-process we apply on the data.

WordPunctTokenizer. WordPunctTokenizer of the NLTK library. It divides a text into a list of words. We chose this tokenizer because it maintains the punctuation marks and separates them from words. In this way, we can exploit the punctuation marks to generate a more accurate stylistic profile of the author.

SnowballStemmer. A stemmer is a tool that removes morphological affixes from a word, reducing it to its stem. A stem is the part of the word that contains no morphological inflections (*love*, *loving* and *loved* are stemmed as *love*). It is part of the NLTK package, and it supports all the languages that are in the PAN dataset, making it suitable for our purpose. Stemming can be very important since by removing the morphological inflectional it permits to recognize that two words are semantically identical even if they differ syntactically.

Convertion with POS tagging. A Part-Of-Speech Tagger is a tool that takes in input a text and assigns a part of speech tag to each word. Some examples of POS tags are *PRP$* that identify a possessive pronoun (my, his, hers), *VB*, that identify a verb at the base form (take), and *VBD* that identify a verb at past tense (took). To create this representation, we firstly tokenize the text, then we use the POS tagger. Finally, we concatenate all the tags and the punctuation marks, adding a space between them. Table 1 show an example of text before and after this process. POS tagging is generally used to underline the structure of the text removing all context-based information. It is useful to identify syntactic patterns in the style of the author. We used the spaCy [1] POS tagger since it handles all the languages present in the dataset, with an accuracy that vary from 95.29% to 97.23%.

Table 1. POS Tagging

Original Text	Text converted with POS tagger
Your eyes opened, scanning the room with slightly dazed wariness. You weren't home, but in a room with grey walls and a glass front. Several cameras were pointed at you. You felt panic rise within sone intro.you	PRP$ NNS VBN , VBG DT NN IN RB VBN NN . PRP VBD RB NN , CC IN DT NN IN JJ NNS CC DT NN NN . JJ NNS VBD VBN IN PRP . PRP VBD JJ NN IN PRP , CC PRP VBD PRP RP IN PRP$ NN CC PRP$ NNS . " NNP , JJ NN

Text distortion. This pre-processing technique for Authorship Attribution task was firstly proposed by [21] and it was used with good results also in [3]. This method consists of masking some part of the text, replacing characters with the '*' symbol. We used this method to maintain only punctuation marks and diacritical characters as shown in Table 2.

5.2 Features

To develop our classifier we tested different kind of features. We heavily use character n-grams due to their robustness in cross-domain settings [13,19,20] and word n-grams. All our features are based on the extraction of sub-sequences of words or characters called n-gram, and then weighting these sequences with the Tf-Idf, where the term frequency is logarithmically scaled. More formally, an n-gram is a contiguous sequence of n items from a given sample of text. The items can be phonemes, syllables, letters

[1] https://spacy.io/

Table 2. Text distortion

Original Text	Text converted with Text Distortion
marqué sur la couverture, avant d'avoir un temps d'arrêt. Le dossier se nommait en effet sobrement « Enterrement de vie de garçon ». Plusieurs souvenirs remontèrent. John sourit doucement en se remém	*****é *** ** **********, ***** *'***** ** ***** *'***ê*. ** ******* ** ******* ** ***** ********* « *********** ** *** ** ***ç** ». ********* ********* ******è****. **** ****** ********* ** ** ***é**

or words. Instead, the Tf-Idf, term frequency-inverse document frequency, is a measure associated to each term, in our case the n-gram sequences, that increases proportionally to the number of times it appears in a document, while it is reduced by the number of documents in the dataset that contain the term. In this way, it is possible to give more importance to features that are frequently used by only one author and less importance to widespread features such as stop-words. With the aim to capture different peculiarities of the author stylistic profile, we tried to combine the benefits of two different approaches: the profile and the instance-based.

Profile-Based Features. In the profile-based approach, we concatenate all the documents that belong to the same author and consider them as a single one. In this way, it is possible to outline the general style of the author. According to this approach, we extract the **Profile** feature. It is the union of Char-gram and Word-gram features computed in the following way. For the char-gram, we take into account the raw text, then we select the char-grams of length between 3 and 5, with a cut-off document frequency less than 0.12. Regarding the word-gram, we firstly tokenize and stem the text. Then we extract the word sequences of length 1 up to 3. We consider only the feature with document frequency strictly lower than 0.3. Once computed both the features we stack them together.

Instance-Based Features. In the instance-based approach, we process and extract the features from each document of an author separately. This last approach allows to extract separate style for each text sample and thus different profiles across different domains. So, for each document, we extract the following features:

- **Char**: from the raw text, we extract char-grams of length 3 up to 5. We select all the sequences with a document frequency of less than 0.12.
- **Dist**: we pre-process the text with the aforementioned distortion technique, then we extract sequences of characters of length between 3 and 5. We use 0.12 as cut-off document frequency value.
- **Stem1&2**: After stemming the text, we extract word uni-grams and bi-grams and for each set of features we weight the terms with the Tf-Idf and discard terms that appear with a document frequency higher than 0.3. Finally, we stack them together.
- **Stem1-3**: We stem the text, and then we extract word-grams of length 1 up 3. We take into account only word-grams that appear in the text with document frequency lower than 0.03.

- **POS**:We transform the text into its POS tagged form then we extract the sequences of tags of length 3 up to 5. We use 0.12 as cut-off document frequency value.

5.3 Classifiers

As the first experiment, we stack all the features mentioned above together, and we test them on different classifiers. We evaluate the performances of the following classifiers: SVM with linear kernel, SVM with RBF kernel, K-nearest neighbors with $K = 3$ and Random Forest. In this experiment, we left all the hyper-parameters to the default values. In Table 3 are shown the results of the classifiers. As we can see, the SVM with linear kernel outperforms other classifiers in almost all the problems. Given this result, we analyze the performance of different kind of features with the linear SVM. In Table 4, we report the score of each kind of feature on different problems. After evaluated these experiments, we try to improve our results with an ensemble classifier that relies on SVMs with linear kernel. The ensemble is a method to combine different classifiers predictions to build a more generally estimator. The ensemble can be based on two methods: *averaging method* and *boosting method*. The averaging method takes the output probabilities of N estimators and combines them usually using average. Instead, the boosting method uses a weak base estimator as a starting point, and then, other weak learners are trained to improve the predecessor. For our final classifier, we build an ensemble architecture based on the averaging method. We combine the predictions of several SVMs with the soft voting function. We finally test different combinations of our features in order to optimize the performances and select only the best features for each language.

In Figure 1, we show our final architecture and features combinations for each language. The continuous line in figure are the features used for all the languages: the **Profile**, **Char**, **Stem1-3** and the **Dist**. While the dashed line represents the feature used only for the English and the Italian classifier **Stem1&2**. Finally, the dotted line depicts the feature we use only for documents written in English and Spanish **POS**. Table 5, resume in tabular form the features used for each language.

5.4 Unknown Detection

In this section, we focus on the process of determining if the author of an unknown document is in the known author set or not. To achieve this goal, we take into account the probabilities results of the first three users. Let's P_1, P_2 and P_3, respectively the probability of the first, the second and third most probable authors. Then we take a decision based on two conditions. The first one is that the difference between P_1 and P_2 must be less than 0.1. The second condition is that the mean of the difference between P_1 and P_2 and P_1 and P_3 must be less 0.7. If both conditions are True, we predict the text as written by an unknown author. Otherwise, we choose the author with the highest probability. The values of 0.1 and 0.7 are based on experimental observations. The idea is that if an author of the known set wrote the unknown document, the difference

Table 3. F1-score for the individual classifier

Problem	SVM Linear	SVM RBF	K-NN	Random Forest
01	78.7	76.9	67.9	68.3
02	57.1	56.2	45.8	42.7
03	71.0	67.1	49.3	45.6
04	45.3	34.2	32.6	33.7
05	53.0	51.8	43.1	44.1
06	65.5	64.3	59.3	43.7
07	62.9	58.0	51.1	38.2
08	64.2	60.7	53.6	39.4
09	73.0	65.0	46.7	43.5
10	54.8	56.7	41.9	33.6
11	69.6	65.9	61.8	60.3
12	64.6	62.3	62.2	48.8
13	72.1	71.7	52.3	55.8
14	78.2	67.6	60.4	71.7
15	70.2	69.5	69.3	65.6
16	87.6	87.1	75.2	68.8
17	71.5	73.4	71.8	40.7
18	82.2	81.9	72.9	68.2
19	66.2	63.7	54.6	38.0
20	52.8	49.3	42.5	26.0
Overall	67.0	64.2	55.7	48.8

Figure 1. Ensemble architecture

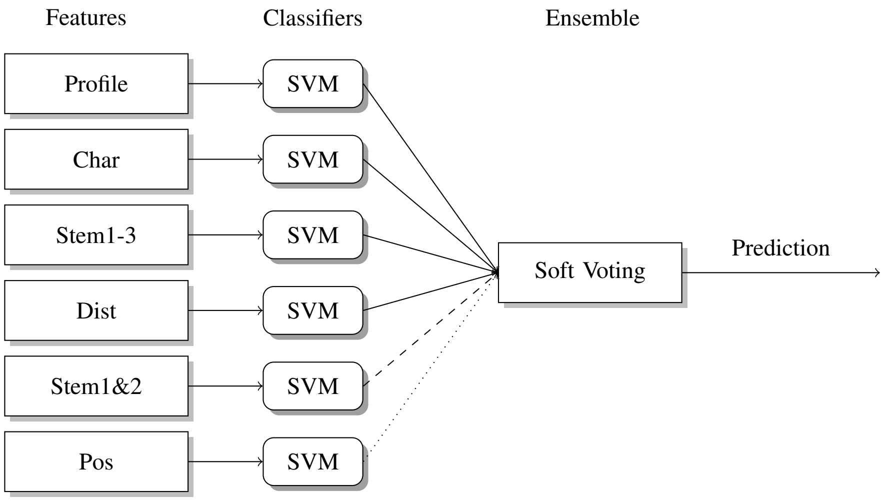


Table 4. F1-score for the individual feature and the aggregation

Problem	Profile	Char	Dist	Stem1-3	Stem1&2	POS
01	76.6	75.4	75.4	76.9	79.0	75.9
02	55.4	55.6	43.9	52.5	53.0	47.2
03	70.8	60.8	41.1	62.5	63.7	45.7
04	48.5	46.2	28.9	43.0	43.1	33.2
05	49.3	53.6	44.4	52.5	49.3	42.5
06	66.4	62.6	63.8	63.0	58.0	19.4
07	57.9	58.4	39.3	57.6	46.8	21.9
08	63.0	58.6	52.2	57.4	45.3	25.3
09	70.1	62.1	52.2	68.6	54.4	21.1
10	54.8	58.5	40.6	59.3	56.6	7.4
11	70.5	64.6	55.6	69.4	71.0	27.1
12	68.3	68.4	54.9	70.6	70.1	20.9
13	72.6	71.9	63.3	68.5	70.5	28.3
14	65.8	51.8	79.4	80.1	81.2	39.4
15	86.4	76.9	56.3	79.9	76.9	19.9
16	84.0	85.0	71.0	81.8	74.9	30.8
17	78.3	75.7	52.3	63.1	61.4	35.1
18	80.6	76.8	61.7	84.1	76.5	30.1
19	67.3	69.4	42.0	69.1	62.7	18.7
20	52.8	53.8	26.1	43.5	50.1	12.2
Overall	67.0	64.3	52.2	65.2	62.2	30.1

Table 5. Table of features for each language

	Profile	Char	Stem1-3	Dist	POS	Stem1&2
English	x	x	x	x	x	x
French	x	x	x	x	-	-
Italian	x	x	x	x	-	x
Spanish	x	x	x	x	x	-

between his probability and the probabilities of the other users must be higher than if we do not have the author of that unknown document.

The idea is that if one of the known authors has written the document, then the distance between their probabilities are high. On the other hand, if the real author is unknown, the probabilities of the known authors are close to each other.

$$Unknown = \begin{cases} True, & P_1 - P_2 < 0.1 \land mean(P_1 - P_2, P_1 - P_3) < 0.7 \\ False, & otherwise \end{cases}$$

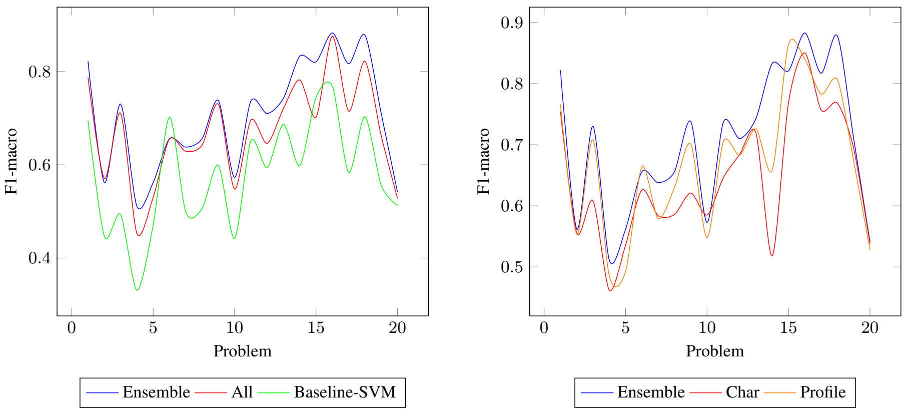
6 Results

PAN organizers chose the *macro-averaged F1 score* as an evaluation metric since the unknown documents to predict are not equally distributed across all the problems. PAN provides to all the participants of the Authorship Attribution task a Train and a Dev dataset as well as a Virtual Private Server (VPS) to deploy the model. The VPS is hosted on the TIRA platform [16], Testbed for Information Retrieval Algorithms. The main function of TIRA is to create a sandbox that the organizers can use to perform the final test and verify the correctness of the result. To better reproduce a real-case scenario, the Test set is unknown to the participants, and the teams involved in the contest can evaluate their model on the Test dataset only on TIRA.

We show in Table 6 the results on the Dev set. As we can see, the overall F1-score is 70.5%, 12.6% higher than baseline-SVM. Looking at the single problems scores, we can see how few problems, 3, 4, and 14, seem to undermine the effectiveness of our model. Problems 16 and 18 are the ones that reach the higher scores, achieving an F1-score of 88.3% and 87.8% respectively. In Figure 2(a), we plot the F1-score of the baseline-SVM, the union of all our features and the final ensemble, for each problem. As we can see, our ensemble method outperforms the baseline-SVM in all but one problem, while it constantly performs better than our classifiers without the ensemble. Looking at the Figure 2(a), it is clear that, even if the methods perform differently, they all struggle with specific problems while achieving better results in others. Further, we plot the F1-score of the two features with the highest results, Profile and Char, used for all languages and the result of our ensemble. As we can see in Figure 2(b) the Profile feature, that is the concatenation of Char and Word n-grams computed with the profile-based approach, performs better than our second best feature (Char), and worse than the ensemble. Looking at the single languages scores we note that the model performs better on the Italian (76.88%) and the Spanish (76.58%) languages, conversely the performance decrease on the English (63.74%) and the French (65.20%) languages. The final result on the Test dataset used by PAN to evaluate the performance of the proposed model, achieve an F1-score of 68%, that is the second-best score in the task, just 1% lower than the first classified.

6.1 Discussion on Classifier Performances

Analyzing our results, we noticed that in the case our classifier assign an author and the author belong to the set of known authors, our classifier have a mean error of only



(a) Comparison between the ensemble, all features stacked together and the baseline-SVM

(b) Comparison between the ensemble, the Profile and the Char features

Figure 2. Comparison between different features set and classifiers

Table 6. F1-score on the Dev dataset

Problem	Baseline-SVM	Baseline-Comp	Ensemble	Delta
01	69.5	68.2	82.2	12.7
02	44.7	33.6	56.2	11.5
03	49.3	50.1	73.0	23.7
04	33.1	49.0	51.1	18.0
05	47.1	34.0	56.2	9.1
06	70.2	69.1	65.6	-4.6
07	49.9	54.2	63.8	13.9
08	50.6	49.2	65.6	15.0
09	59.9	60.8	73.8	13.9
10	44.2	50.1	57.3	13.1
11	65.1	59.5	73.7	8.6
12	59.4	50.8	71.0	11.6
13	68.7	73.1	74.3	5.6
14	59.8	78.0	83.3	23.5
15	74.5	71.2	82.1	7.6
16	76.8	70.5	88.3	11.5
17	58.4	62.3	81.7	23.3
18	70.3	65.9	87.8	17.5
19	55.6	40.3	71.0	15.4
20	51.3	22.3	54.1	2.8
Overall	57.9	55.6	70.5	12.6

2.95%. On the other hand, in the case in which in the classification is involved the unknown detector we have a mean error of 26.17%. So, we further investigate about the performance of our classifier in the closed-set scenario. In other words, we run our Authorship Attribution model only on the documents that are written by known author in the Dev dataset. Then, we label as the right author, the one with highest probability score. After the execution we score an overall accuracy of 87% on a total of 2,646 documents. More in details we achieve a single accuracy of 90% on English, 82.4% on French, 84.3% on Italian and 88.5% on Spanish. To better understand the impact of the unknowns detector on the model, we repeat the experiment on a fake open-set scenario. In this scenario, we take into account the possibility that there are unknown authors, but we use the same dataset of the previous experiment where they are not. This time we achieve as overall result an accuracy of 78.7%, that is 8.3% lower of the experiment in the closed-set scenario. This results show that in the absence of unknown author (i.e., in a closed-set scenario), our classifier achieves excellent results. However, when we move on the open-set the unknowns detector induces an error of 8.3%. This drop in performance is pretty normal, and it is well known that the Authorship Attribution in open-set scenario is more difficult then in a closed-set. Nonetheless, the results clearly indicate that although our methodology to detect the unknown authors performs slightly better than the baseline, further improvements are needed.

7 Conclusion and Future Work

In this paper, we proposed our solution for the 2019 Authorship Attribution PAN task. We present a model that relies on different classifiers fitted with a single feature rather than more features for a single classifier. We use an ensemble approach to combine all the probabilities of our single classifiers for each language and increase their result. We use different pre-processing techniques to extract features of a different meaning. We use text distortion, tokenization, stemming, and POS tagging to prepare the text for the extraction. To solve the problem of the unknown authors, we introduced a method that takes into account the three most similar author for the disputed text, instead of only the first two. With our approach, we outperform the baseline for almost every problem.

Analyzing our result on different problems, we notice that our performances tend to decrease in the presence of a high number of author missing in the training data. So, we believe that improving the algorithm to detect when an author is unknown could lead to better results in this problem and hence a better result in the overall score. Looking at the baseline in Table 6, we notice that some times the compression method seems to reach high results. It could be useful to understand what kind of pattern the compression identify and use it in order to improve our classifier.

Acknowledgment

This work was supported in part by the MIUR under grant “Dipartimenti di eccellenza 2018-2022" of the Department of Computer Science of Sapienza University.

References

1. Abbasi, A., Chen, H.: Writeprints: A stylometric approach to identity-level identification and similarity detection in cyberspace. ACM Transactions on Information Systems (TOIS) 26(2), 7 (2008)
2. Brocardo, M.L., Traore, I., Woungang, I., Obaidat, M.S.: Authorship verification using deep belief network systems. International Journal of Communication Systems 30(12), e3259 (2017)
3. Custódio, J.E., Paraboni, I.: Each-usp ensemble cross-domain authorship attribution. Working Notes Papers of the CLEF (2018)
4. Daelemans, W., Kestemont, M., Manjavancas, E., Potthast, M., Rangel, F., Rosso, P., Specht, G., Stamatatos, E., Stein, B., Tschuggnall, M., Wiegmann, M., Zangerle, E.: Overview of PAN 2019: Author Profiling, Celebrity Profiling, Cross-domain Authorship Attribution and Style Change Detection. In: Crestani, F., Braschler, M., Savoy, J., Rauber, A., Müller, H., Losada, D., Heinatz, G., Cappellato, L., Ferro, N. (eds.) Proceedings of the Tenth International Conference of the CLEF Association (CLEF 2019). Springer (Sep 2019)
5. Halvani, O., Graner, L.: Author clustering using compression-based dissimilarity scores
6. Halvani, O., Graner, L.: Cross-domain authorship attribution based on compression
7. Kešelj, V., Peng, F., Cercone, N., Thomas, C.: N-gram-based author profiles for authorship attribution. In: Proceedings of the conference pacific association for computational linguistics, PACLING. vol. 3, pp. 255–264. sn (2003)
8. Kestemont, M., Luyckx, K., Daelemans, W., Crombez, T.: Cross-genre authorship verification using unmasking. English Studies 93(3), 340–356 (2012)
9. Kestemont, M., Stamatatos, E., Manjavacas, E., Daelemans, W., Potthast, M., Stein, B.: Overview of the Cross-domain Authorship Attribution Task at PAN 2019. In: Cappellato, L., Ferro, N., Losada, D., Müller, H. (eds.) CLEF 2019 Labs and Workshops, Notebook Papers. CEUR-WS.org (Sep 2019)
10. 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)
11. Koppel, M., Schler, J., Bonchek-Dokow, E.: Measuring differentiability: Unmasking pseudonymous authors. Journal of Machine Learning Research 8(Jun), 1261–1276 (2007)
12. Leonard, R.A., Ford, J.E., Christensen, T.K.: Forensic linguistics: Applying the science of linguistics to issues of the law. Hofstra L. Rev. 45, 881 (2016)
13. Luyckx, K., Daelemans, W.: The effect of author set size and data size in authorship attribution. Literary and linguistic Computing 26(1), 35–55 (2011)
14. Oliveira Jr, W., Justino, E., Oliveira, L.S.: Comparing compression models for authorship attribution. Forensic science international 228(1-3), 100–104 (2013)
15. Potha, N., Stamatatos, E.: A profile-based method for authorship verification. In: Hellenic Conference on Artificial Intelligence. pp. 313–326. Springer (2014)
16. Potthast, M., Gollub, T., Wiegmann, M., Stein, B.: TIRA Integrated Research Architecture. In: Ferro, N., Peters, C. (eds.) Information Retrieval Evaluation in a Changing World - Lessons Learned from 20 Years of CLEF. Springer (2019)
17. Ruder, S., Ghaffari, P., Breslin, J.G.: Character-level and multi-channel convolutional neural networks for large-scale authorship attribution. arXiv preprint arXiv:1609.06686 (2016)
18. Shrestha, P., Sierra, S., Gonzalez, F., Montes, M., Rosso, P., Solorio, T.: Convolutional neural networks for authorship attribution of short texts. In: Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers. vol. 2, pp. 669–674 (2017)
19. Stamatatos, E.: A survey of modern authorship attribution methods. Journal of the American Society for information Science and Technology 60(3), 538–556 (2009)

20. Stamatatos, E.: On the robustness of authorship attribution based on character n-gram features. Journal of Law and Policy 21(2), 421–439 (2013)
21. Stamatatos, E.: Authorship attribution using text distortion. In: Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers. pp. 1138–1149 (2017)