Cross-domain Authorship Attribution: Author Identification using a Multi-Aspect Ensemble Approach

Notebook for PAN at CLEF 2019

Mostafa Rahgouy[1], Hamed Babaei Giglou[1] , Taher Rahgooy[2], Mohammad Karami Sheykhlan[1] , and Erfan Mohammadzadeh [1]

[1] University of Mohaghegh Ardabili, Computer Science Department, Ardabil, Iran
mostafarahgouy@student.uma.ac.ir
{hamedbabaeigiglou, mohammadkaramisheykhlan, er.mohammadzadeh}@gmail.com
[2] Tulane University , Computer Science Department, New Orleans, LA, USA
trahgooy@tulane.edu

Abstract Author Attribution (AA) as one of the most important tasks of authorship analysis attracted huge body of research in recent years. In this task, given a document, the goal is to identify its author from a set of known authors and samples of their writings. In PAN 2019 shared tasks, the AA task is expanded in two ways. First, by having documents written by authors other than the known authors (UNK documents). Second, using a cross-domain set of documents. The task baseline and previous works mainly rely on character-level representation of documents because of their better generalization capability across different languages and domains. However, we hypothesize that ignoring coarse-grain features of documents may lead to loss of valuable information about the author's style. In this paper we propose an ensemble approach that combines models built upon different levels of document representation in order to investigate this hypothesis. Experimental results presented in this paper show that the coarse-grained representations of documents play an important role in identifying the authors style alongside the fine-grained representations.

Keywords: Authorship Attribution, Author Identification, Natural Language Processing, Supervised Machine Learning, Stacking ensemble.

1 Introduction

In recent years, a significant amount of research has attended to formulation, modeling, and evaluation of authorship related tasks such as author identification [15,6],

author profiling [15], and author obfuscation [14] which have many practical applications in electronic commerce, forensics, and humanities research [7,3].

In PAN 2019 shared tasks the Authorship Attribution(AA) task [5] deals with the problem of identifying a document's author from a set of candidate authors. This task could be considered as a special case of text classification [18] where candidate authors serve as target classes. AA problem has a long history in natural language processing(NLP) which dates back to the 19th century [10]. Even after more than ten decades, the problem is still far from being solved and has become an important research subject, across may fields and domains. However, the majority of existing research focuses on closed-set which assumes that the candidate set is closed and thus contains sample writings of true author of the unknown document, but at PAN 2019 [5] the focus is on open-set AA which considers a more realistic setting, where the true author is no longer believed to be present in the candidate set. It goes without saying open-set case makes the problem more challenging than the AA task at PAN 2018 [6].

The rest of the paper is organized as follows. Section 2 provides background and presents some related works on text classification in general. Section 3 introduce our AA approach. Results are covered in Section 4. In Section 5 we draw a conclusion and finally we suggests ideas for future research in section 6.

2 Related Work

The author attribution task is presented for the first time in PAN 2011 [1] in a single-domain setting and continued to be part of PAN afterwards. AA task has been expanded and modified to more complicated and challenging tasks by considering cross-domain, multi-language settings, and open-set author candidates settings [5].

In [17] it is shown that cross-domain setting far more challenging than the single-domain setting and increasing variety of topics in the training set plays a pivotal role in the ability of their model to learn the task. Character n-grams are used in many works as main representation of the documents and shown to be efficient and robust [9,19,20].

In a similar work to our work, author of [4] make use of an ensemble approach with standard character n-grams, character n-grams with non-diacritic distortion and word n-grams. The work focuses on the use of character-level information, whereas our proposal will focus on both word-level and character-level information. In another related work [12] make use of traditional character n-gram analysis in combination with a linear SVM as a classifier. They found the optimal values for their model with dynamic and ad-hoc grid search approach and achieved reasonable results at PAN 2018.

3 Proposed Approach

In this section we present our proposed model. We hypothesize that a combination of character-level and word-level representation of document could work as complimentary set of information and hence improve the resulting predictions. Therefore, we used an ensemble of classifiers, each trained on one set of representation. In the following we present the details of our approach.

3.1 Data Preprocessing

The first step in the proposed algorithm is to pre-process the input documents. In this step, we removed the punctuations from documents and then we split the documents to words using *WordPunctTokenizer* of NLTK 3.0 Toolkit[2]. Next, we removed stop-words from tokens and stemming words using *PorterStemmer*. The final preprocessed documents are used to feed TF-IDF and Word2Vec models.

3.2 Data Representation

In this section we described three models for the stacking ensemble.

N-gram The baseline model provided by PAN 2019 authorship attribution shared task [5] uses character 3-gram frequencies in combination with a linear SVM. We used this model and fine-tuned its parameters using grid search implemented in scikit-learn library [13]. The obtained optimized values are listed in Table 1. All runs were performed with 5-fold cross validation, while optimizing the F1-Macro target.

Module	Parameters	Possible values
Feature Extraction	minimal document frequency	3, **5**
	n-gram order	3, **4** ,5
	lowercase	true, **false**
	script accents	true , **false**
Transformation	Scaling	None , **MaxAbsScaler**
Classifier	C parameter of SVM	**0.01**, 0.1, 10, 100
	SVM kernel	**linear** , rbf

Table 1. Parameter tuning using grid search. Bold values indicate the parameter values that resulting best F1-Macro.

TF-IDF Term Frequency-Inverse Document Frequency (TF-IDF) [16] is a common feature transformation technique for detecting authors style. First, we build a vocabulary using pre-processed train-set for each problem with frequency term **5**. Next, using the scikit-learn's *TfidfVectorizer* method we convert a collection of raw documents to a matrix of TF-IDF features. We train 4 different classifiers using a TF-IDF to find the best classifier with TF-IDF features. Table 2 shows the Macro-Averaged F1 results of this experiment on the train and development set. Based on this experiment we chose the *LinearSVC* for a TF-IDF features classifier.

Word Embedding In order to choose between word embedding *Doc2Vec* [8] and *Word2Vec* [11], we set an experiment with different classifiers to chose the best word embedding. Table 3 shows that Word2Vec obtained the best results most of the time. So based on these experiments, we chose the *Word2Vec* as our third feature extractor and *LogisticRegression* as classifier. We trained *Word2Vec* and *Doc2Vec* for each problem separately using the all the texts provided in that problem.

Classifier	Macro-Averaged F1
LogisticRegression	0.4370
LinearSVC	**0.4626**
BernoulliNB	0.4388
MLPClassifier	0.4601

Table 2. Model selection for TF-IDF features. Averaged F1-Macro for classifiers trained on TF-IDF features.

Classifier + Word Embedding	Macro-Averaged F1
LogisticRegression + Word2Vec	**0.4338**
LogisticRegression + Doc2Vec	0.1811
LinearSVC + Word2Vec	0.4239
LinearSVC + Doc2Vec	0.1946
MLPClassifier + Word2Vec	0.4278
MLPClassifier + Doc2Vec	0.3742
BernoulliNB + Word2Vec	0.3425
BernoulliNB + Doc2Vec	0.3462

Table 3. Model selection for word embedding features. Averaged F1-score for Doc2Vec and Word2Vec features.

3.3 Ensemble

There are many different types of ensembles; stacking is one of them. It is one of the more general types and can theoretically represent any other ensemble technique. Stacking involves training a learning algorithm to combine the predictions of several other learning algorithms. We use one of the simplest forms of Stacking, which we train three different classifier described in Section 3.2. We used *CalibratedClassifierCV* from scikit-learn library to compute the likelihood of each candidate in each classifier for a test example. Next, we calculate the average outputs of models in the ensemble to make the final prediction. For a given sample, we select the candidate with the highest probability as the output if the difference between the most probable and second most probable prediction probability is bigger than a threshold otherwise we predict it as a <*UNK*>. This process is visualized in Figure 1.

Table 4 shows examples of three documents of our stacking ensemble and detailed likelihoods obtained from each classifier for all 9 candidates in one of the problems in the train set. Based on this table TF-IDF and Word2Vec support the N-gram approach and in other places improve the N-gram predictions and it also in some cased the N-gram model fixed the TF-IDF and Word2Vec wrong predictions.

3.4 Threshold for *UNK*

We experimented with different UNK thresholds for the proposed ensemble which are presented in Table 5. Based on these results we choose best threshold **0.08** with averaged F1-macro **0.61875**.

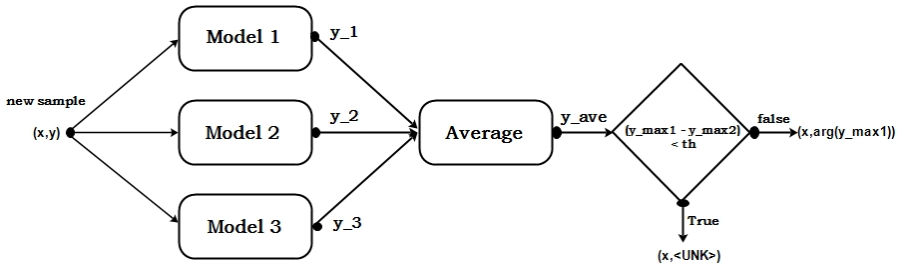

Figure 1. The architecture of the stacking ensemble. *th* is the pre-defined threshold and is **0.08** .

#	Models	P(ca1)	P(ca2)	P(ca3)	P(ca4)	P(ca5)	P(ca6)	P(ca7)	P(ca8)	P(ca9)	Pred
1	tfidf	0.219	0.047	0.033	0.201	0.093	0.080	**0.235**	0.042	0.045	ca7
	ngram	0.068	0.056	0.052	0.042	0.125	0.094	**0.451**	0.027	0.080	ca7
	word2vec	0.121	0.096	0.072	0.101	**0.184**	0.132	0.167	0.073	0.050	ca5
	ensemble	0.136	0.066	0.052	0.115	0.134	0.102	**0.285**	0.047	0.059	**ca7**
15	tfidf	0.122	0.010	0.030	0.083	0.162	**0.406**	0.008	0.103	0.071	ca6
	ngram	0.102	0.009	0.027	0.062	**0.320**	0.316	0.025	0.0443	0.090	ca5
	word2vec	0.146	0.018	0.034	0.153	0.153	**0.337**	0.021	0.054	0.080	ca6
	ensemble	0.124	0.012	0.030	0.099	0.212	**0.353**	0.018	0.067	0.081	**ca6**
92	tfidf	0.129	0.028	0.044	0.255	**0.256**	0.165	0.072	0.015	0.030	ca5
	ngram	0.191	0.053	0.050	**0.430**	0.057	0.047	0.065	0.092	0.011	ca4
	word2vec	0.131	0.132	0.091	0.188	0.079	**0.197**	0.082	0.049	0.047	ca6
	ensemble	0.151	0.071	0.061	**0.291**	0.131	0.136	0.073	0.052	0.029	**ca4**

Table 4. Examples of stacking ensemble. Documents belong to *problem00001*.

3.5 Other Features

In the process of feature engineering, we explored many other ideas, some of which performed poorly and thus we did not get them involved to our final approach. Yet, we feel some of them are worth mentioning.

Contracted Word-forms we used is based on the discrepancies in spelling for words that allow contracted forms, e.g.,I will (I'LL), are not (aren't). People typically favor one of the alternatives, and thus we use forms based on contracted apostrophes as discriminative features for detecting the style of each author.

Quotation Marks Some authors may prefer either single or double quotation marks. We use the difference between the number of single and double quotes in a given document.

Sentence Length we noticed that some authors prefer to write long sentences where they use more conjunction in their text whereas some of them use short sentences as a result we consider length as a feature also we calculate the number of conjunctions in a given document and treat them as a feature.

Negations Another feature we use is based on the negation form e.g., impossible(not possible) to identifying the similarity of authors. We calculated the number of

Threshold	Macro-Averaged F1
0.05	0.60976
0.08	**0.61875**
0.1	0.61366
0.15	0.55659
0.2	0.45940

Table 5. Averaged F1-macro for different UNK thresholds.

negations for each given document and added them to other features but we didn't get any notable result.

4 Experimental Results

In Table 6, the performance of all possible combination of three models described in section 3 are presented. These results approve our hypothesis that coarse-grained representations complement the fine-grained representations of documents. Also, it is clear that the ensemble of all three models is the best model in average.

Language	Models						
	N-gram	Word2vec	TF-IDF	N-gram+ Word2vec	N-gram+ TF-IDF	Word2vec+ TF-IDF	All three
EN	0.49	0.32	0.41	0.53	**0.54**	0.41	0.53
FR	0.55	0.47	0.33	**0.58**	0.54	0.50	**0.58**
IT	0.46	0.55	0.48	0.65	0.68	0.56	**0.69**
SP	0.63	0.53	0.51	0.67	0.66	0.58	**0.68**

Table 6. Averaged F1-macro results on development dataset for each different combination of models. The results are separated for each language

Furthermore, we present the detailed results of the ensemble model on development dataset of PAN 2019 in Table 7. As you can see, the development set includes **4051** unknown documents and composed of 20 problems divided in **four** languages (five problems each). The overall score obtained by this model is **0.61875** .

5 Conclusion

In this paper we proposed a model for Cross-domain Authorship Attribution task in PAN 2019. We presented our approach, which uses a TF-IDF, Word2Vec and N-grams representation of document to train three type of models and make predictions using an ensemble of those models. Next, we tuned the models and ensemble parameters using

Problem#	Language	Macro-Averaged F1	TP	Test Size
Problem01	en	0.63267	412	561
Problem02	en	0.50442	71	137
Problem03	en	0.49664	119	211
Problem04	en	0.47285	142	273
Problem05	en	0.54606	183	264
Problem06	fr	0.63153	82	121
Problem07	fr	0.53467	61	92
Problem08	fr	0.60916	278	430
Problem09	fr	0.54458	133	239
Problem10	fr	0.58112	23	38
Problem11	it	0.59091	75	139
Problem12	it	0.66867	82	116
Problem13	it	0.72099	138	196
Problem14	it	0.62285	36	46
Problem15	it	0.86705	46	54
Problem16	sp	0.72977	117	164
Problem17	sp	0.74521	82	112
Problem18	sp	0.73916	175	238
Problem19	sp	0.57426	275	450
Problem20	sp	0.56253	98	170
Overall score		**0.61875**	**2628**	**4051**

Table 7. Detailed averaged F1-macro results of the ensemble on development dataset.

an ad-hoc grid search approach to find the optimal values. Our evaluation shows that our approach is very capable of distinguishing authors from the others. The proposed algorithm implemented in *Python* and published on *GitHub*[1].

References

1. Argamon, S., Juola, P.: Overview of the International Authorship Identification Competition at PAN-2011. In: Petras, V., Forner, P., Clough, P. (eds.) Notebook Papers of CLEF 2011 Labs and Workshops, 19-22 September, Amsterdam, Netherlands. CEUR-WS.org (Sep 2011), http://www.clef-initiative.eu/publication/working-notes
2. Bird, S., Klein, E., Loper, E.: Natural language processing with Python: analyzing text with the natural language toolkit. " O'Reilly Media, Inc." (2009)
3. Coulthard, M.: On admissible linguistic evidence. JL & Pol'y 21, 441 (2012)
4. Custódio, J., Paraboni, I.: EACH-USP Ensemble Cross-Domain Authorship Attribution—Notebook for PAN at CLEF 2018. In: Cappellato, L., Ferro, N., Nie, J.Y., Soulier, L. (eds.) CLEF 2018 Evaluation Labs and Workshop – Working Notes Papers, 10-14 September, Avignon, France. CEUR-WS.org (Sep 2018)
5. 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

[1] https://github.com/HamedBabaei/PAN2019_cross_domain

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)
6. Kestemont, M., Tschuggnall, M., Stamatatos, E., Daelemans, W., Specht, G., Stein, B., Potthast, M.: Overview of the author identification task at pan-2018: cross-domain authorship attribution and style change detection. In: Working Notes Papers of the CLEF 2018 Evaluation Labs. Avignon, France, September 10-14, 2018/Cappellato, Linda [edit.]; et al. pp. 1–25 (2018)
7. Koppel, M., Seidman, S.: Automatically identifying pseudepigraphic texts. In: Proceedings of the 2013 conference on empirical methods in natural language processing. pp. 1449–1454 (2013)
8. Le, Q.V., Mikolov, T.: Distributed representations of sentences and documents. In: ICML (2014)
9. 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)
10. Mendenhall, T.C.: The characteristic curves of composition. Science 9(214), 237–249 (1887)
11. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781 (2013)
12. Murauer, B., Tschuggnall, M., Specht, G.: Dynamic Parameter Search for Cross-Domain Authorship Attribution—Notebook for PAN at CLEF 2018. In: Cappellato, L., Ferro, N., Nie, J.Y., Soulier, L. (eds.) CLEF 2018 Evaluation Labs and Workshop – Working Notes Papers, 10-14 September, Avignon, France. CEUR-WS.org (Sep 2018)
13. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: Machine learning in Python. Journal of Machine Learning Research 12, 2825–2830 (2011)
14. Potthast, M., Schremmer, F., Hagen, M., Stein, B.: Overview of the Author Obfuscation Task at PAN 2018: A New Approach to Measuring Safety. In: Cappellato, L., Ferro, N., Nie, J.Y., Soulier, L. (eds.) Working Notes Papers of the CLEF 2018 Evaluation Labs. CEUR Workshop Proceedings, CLEF and CEUR-WS.org (Sep 2018)
15. Rangel Pardo, F., Montes-y-Gómez, M., Potthast, M., Stein, B.: Overview of the 6th Author Profiling Task at PAN 2018: Cross-domain Authorship Attribution and Style Change Detection. In: Cappellato, L., Ferro, N., Nie, J.Y., Soulier, L. (eds.) CLEF 2018 Evaluation Labs and Workshop – Working Notes Papers, 10-14 September, Avignon, France. CEUR-WS.org (Sep 2018)
16. Salton, G., McGill, M.J.: Introduction to modern information retrieval (1986)
17. Sapkota, U., Solorio, T., Montes, M., Bethard, S., Rosso, P.: Cross-topic authorship attribution: Will out-of-topic data help? In: Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers. pp. 1228–1237 (2014)
18. Sebastiani, F.: Machine learning in automated text categorization. ACM computing surveys (CSUR) 34(1), 1–47 (2002)
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 & Policy 21(2) (2013)