Authorship Attribution in Fan-Fictional Texts given variable length Character and Word N-Grams

Notebook for PAN at CLEF 2019

Lukas Muttenthaler*,[1] [0000−0002−0804−4687], Gordon Lucas[2] [0000−0002−5626−6890], and Janek Amann[0000−0002−9868−9568]

University of Copenhagen (UCPH)
[1]Department of Computer Science, [2]Department of Psychology
mnd926@alumni. ku.dk

Abstract The task of authorship attribution (AA) requires text features to be represented according to rigorous experiments. In the current study, we aimed to develop three different n-gram models to identify authors of various fan-fictional texts. Each of the three models was developed as a variable-length n-gram model. We implemented both a standard character n-gram model ($2-5$ gram), a distorted character n-gram model ($1-3$ gram) and a word n-gram model ($1-3$ gram) to not only capture the syntactic features, but also the lexical features and content of a given text. Token weighting was performed through term-frequency inverse-document frequency (tf-idf) computation. For each of the three models, we implemented a linear Support Vector Machine (SVM) classifier, and in the end applied a soft voting procedure to take the average of the classifiers' results. Results showed, that among the three individual models, the standard character n-gram model performed best. However, the combination of all three classifier's predictions yielded the best results overall. To enhance computational efficiency, we computed dimensionality reduction using Singular Value Decomposition (SVD) before fitting the SVMs with training data. With a run time of approximately 180 seconds for all 20 problems, we achieved a macro F1-score of 70.5% for the development corpus and a F1-score of 69% for the competition's test corpus, which significantly outperformed the PAN 2019 baseline classifier. Thus, we have shown that it is not a single feature representation that will yield accurate classifications, but rather the combination of various text representations that will depict an author's writing style most thoroughly.

Keywords: authorship attribution, n-grams, tf-idf, Support Vector Machine, Singular Value Decomposition

* *Corresponding author.*

1 Introduction

Authorship Attribution (AA) is the task of determining the author of a text from a set of candidates. It can be a daunting exercise for both humans and machines, if one does not know which parts of a document represent an author's writing style. However, if features are represented according to rigorous experiments (e.g., through the use of regular expressions and hyper parameter optimization) and adequately capture the syntactic use of language, it may well support automated systems which aim to recognize a text's author. In the context of machine learning, AA can be regarded as a multi-class, single-label text classification problem [6]. Its applications include plagiarism detection and forensic linguistics as well as research in literature [4,11]. For comprehensive surveys of this topic see [5] and [12].

In this working notes paper, we describe our approach to the cross-domain AA task of the PAN 2019 competition, which comprised of the identification of the authors of fan fiction [8]. Fan fiction describes literary works written by fans based on a previous, original literary work (also called the *fandom*). Fan fiction usually includes the characters from the original, it does, however, change or reinterpret other parts of the story, such as settings or endings, or explores a less prominent character in more detail [7]. In recent years, fan fiction has generated some controversy concerning the violation of intellectual rights [8]. In this years PAN competition, one of the tasks constitutes of 20 author identification problems in English, French, Italian and Spanish (5 for each language). Each problem featured 9 candidate authors, for each of which 7 known texts were provided. The term cross-domain in this particular context refers to the fact that the texts with known authorship were from different fandoms, whereas the unknown texts were in a single and different fandom [8]. While PAN 2018 featured a closed set of candidate authors, this years task presents an open set problem: The real author of some tests cases are unknown.

2 Method

2.1 Feature Representation

Similarly to last year's best performing team [2], we deployed three different n-gram models to represent the fan-fictional texts. We implemented both character and word n-gram models. All models are variable length n-gram models as, according to recent experiments, variable length n-gram models both represent an author's style more adequately and yield higher precision and recall scores than fixed n-gram models which do not capture the full scope of syntactic and lexical features [2,3,6].

The first model we implemented, which was a standard character n-gram model, consisted of bigram, trigram, four-gram and five-gram token representations (i.e., $2-5$ grams). Hyperparameter tuning experiments revealed that, $2-5$ gram models yield the best results and represent an author's text more thoroughly than any other additive lower or higher n-gram text representations. However, for this standard character model we also computed an additional vector of unigram punctuation marks, which we then concatenated with all other character n-gram representations after pre-processing computation. We kept punctuation marks as we consid-

ered them important features for an author's use of language. The use, and in particular the frequency, of punctuation marks in a given text, reveals crucial information about an author's writing style [1,4]. Furthermore, we converted each non-zero digit into a 0, since it is not a digit that represents a unique writing style, but rather how numbers are depicted in a given text (e.g., $1k$ vs. 1000; $1,000$ vs. 1.000). The latter step was implemented for all three models. This particular model primarily captured syntactic and morphological features.

Table 1: Showcase of text distortion. The text was extracted from problem set 2.

Original Text	**Distorted Text**
"Yes," she says, "the needs of the many..."	"***," *** ****, "*** ***** ** *** ****..."
"Yes," you agree. "	"***," *** *****. "** ***** ** *** ***** ***** **
It would be the worst thing in my life, but	** ****, *** ** ***** ** ***** **. **** ** * *****'*
it would be worth it. Even if I weren't able	**** ** ****** **, *** ********
to choose it, for whatever	

The second model, which may be deemed a distorted character model, consisted of unigram, bigram and trigram token representations (i.e., $1-3$ grams). This model particularly draws attention to punctuation marks, spacing, diacritics, and all other non-alphabetical characters, including numbers [2]. We replaced all non-diacritics through an asterisk symbol (*) to yield a uniform representation of dispensable characters and hence distorts the text (see Table 1). We deployed this model to account for an author's orthography and punctuation that is not necessarily dependent on other aspects of language use. We suspected that punctuation and orthography might well represent individual features of language. To provide an example, an author might have preferences for ellipses, or might leave out accents in Spanish or French texts.

Lastly, we computed a variable-length word n-gram model, which consisted of unigram, bigram and trigram token representations (i.e., $1-3$ grams). As a word n-gram model, contrary to a character n-gram model, rather captures the vocabulary and register than morhological and syntactic features and thus draws attention to the content, we removed punctuation marks and special characters as such tokens do not reveal decisive information about an author's choice of words. We only kept alphanumerical characters.

2.2 Text Preprocessing

To summarize, for all three n-gram models, we computed the following text preprocessing steps:

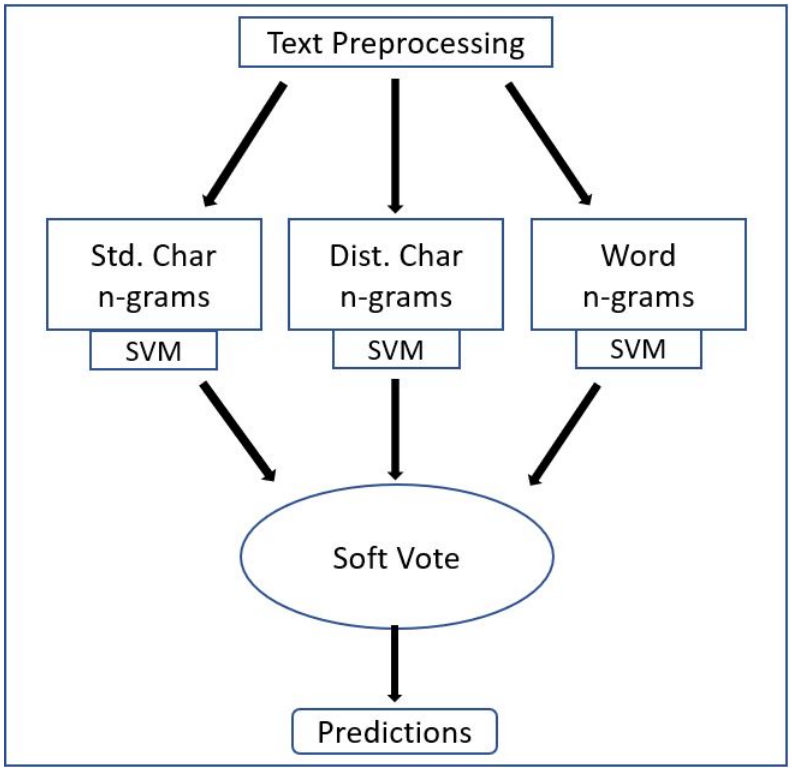
– Each **non-zero digit** was represented by a 0 digit place-holder. This step was computed, since different digits neither reveal crucial information about the vocabulary and register nor about the syntax of a given text. What's important is whether a digit or number appeared in a text and how frequently it was used. Furthermore, to gather information about syntactic features, it is indispensable

to know how numerical characters were depicted by an author (see section on standard character n-gram model).
- Each **hyperlink**, that appeared in a given text, was replaced by an **@** sign. We computed the latter step, as we did not deem hyperlinks crucial features for an author's use of language. What's decisive is whether and how frequently hyperlinks were used within a given text but not the actual hyperlink that was cited.
- For each string, new lines denoted as either **"\n"** or **"\t"** were replaced by the empty string.
- We did **not** perform text **lower-casing** as we suppose, that the usage (and in particular its corresponding frequency) of upper-case letters (e.g., nouns, proper nouns, entities, named entities) reveal important information about the writing style of an author.
- We did not remove stop words as we believe, that **function words** (e.g., "the", "a", "have") - which, in the vast majority of documents, are the most frequently used words - reveal decisive insights into an author's use of language. However, we still assign lower weights to terms that appear highly frequently across the entire corpus.
- Weighting for both character and word n-gram models was performed through term-frequency inverse-document-frequency (**tf-idf**) computation to assign lower weights to words, which appear more frequently across the entire corpus, and higher weights to words which are found only in specific documents and thus reveal crucial information about a text. To prevent a division by zero, we computed **smooth inverse document frequency** instead of normal idf. Smoothing idf means adding a 1 to the denominator within the logarithmic fraction: $\log\left(\frac{N}{1+n_t}\right)$.
- To decrease dimensionality, enhance computational efficiency and reduce time complexity, we computed **Singular Value Decomposition** (SVD) for each of the three variable length n-gram models. The final number of dimensions chosen ($d = 63$) was dependent on the number of samples per problem. As each problem consisted of 63 texts and d cannot be higher than the number of samples in SVD, we deemed this number as an appropriate size for d to capture the most variance.

2.3 Models

For each of the three models (*standard character, distorted character and word n-grams*) we developed linear, multi-class Support Vector Machines (SVM). In so doing, we made use of Python's Sklearn library [9]. Each classifier was cross-validated three times.

Results of the three individual SVMs were combined in a soft-voting manner: We simply averaged the probabilities for each candidate author across the three models. Initial experiments revealed that a pure averaging yields better results than feeding the probabilities into a new classifier. Hence, we did not deploy a fourth model (see Figure 1).



The soft voting computation was performed as follows:

$$\arg\max \frac{\sum_{i=1}^{k} p_i}{\mathrm{k}} \tag{1}$$

where p_i is the probabilities vector for each individual model and $k = 3$

Figure 1: Illustration of the model architecture. After text pre-processing, features for the individual models were extracted. The three individual probability outputs are combined through a soft voting procedure into a final prediction.

The procedure was as follows:

- Firstly, the predicted probabilities obtained by each individual classifier were concatenated into a $n \times d$ matrix, where n is the number of classifiers ($n = 3$) and d denotes the number of classes / candidates.
- Secondly, for each text, we computed the average among the three $1 \times d$ probability vectors, which, according to soft voting, yields a more accurate probabilities depiction than individual probability distributions.
- Lastly, for each text, a "new" average probability distribution was computed, which served as the probabilities vector for the final prediction.

Our classifiers were required to consider that a test text may have been written by an unknown author. To enhance this algorithmic decision-making process, our models classified a text's author as unknown, if the difference between the model's highest and second highest probabilities was below 0.1, or if the highest probability was below 0.25. The latter served as an additional feature (compared to the PAN 2019 baseline model), we regarded as crucial to include in our algorithm. The hyper parameters of the final analysis pipeline are summarized in Table 2. Our final model for the PAN 2019 shared task was deployed on a virtual machine using the TIRA architecture [10].

Table 2: Settings of the final model

term extraction	n-gram range	std_char (2-5) dist_char (1-3) word (1-3)
feature extraction	tf	sublinear
	idf	smoothed
	norm	L2
	proportion of n-grams used	0.5
scaling	MaxAbsScaler	
dimensionality reduction	SVD	63 components
classification	classifier	SVM
	decision procedure	soft vote
	metric	average
	min difference	0.1
	min probability	0.25

3 Results

Table 3 summarizes our results obtained for the development corpus. We compared performances between the official PAN 2019 baseline SVM classifier, our three individual classifiers and the soft vote average model. The highest scores for each problem are displayed in bold face.

Table 3: Comparison of Macro F1 scores for our different models on the PAN 2019 AA development corpus

Problem	Language	nr test texts	baseline	char	dist	word	soft vote
01	en	561	0.695	0.741	0.742	0.631	**0.857**
02	en	137	0.447	0.552	0.423	0.455	**0.553**
03	en	211	0.491	0.620	0.579	0.489	**0.738**
04	en	273	0.331	0.417	0.238	0.299	**0.537**
05	en	264	0.473	0.481	0.417	0.475	**0.585**
06	fr	121	0.702	0.711	0.655	0.437	**0.777**
07	fr	92	0.499	0.551	0.427	0.469	**0.588**
08	fr	430	0.506	0.569	0.474	0.411	**0.673**
09	fr	239	0.599	0.656	0.636	0.437	**0.723**
10	fr	38	0.442	0.544	0.481	0.303	**0.658**
11	it	139	0.651	0.708	0.662	0.505	**0.780**
12	it	116	0.594	**0.685**	0.527	0.584	0.658
13	it	196	0.687	0.762	0.572	0.625	**0.786**
14	it	46	0.583	0.680	0.725	0.464	**0.750**
15	it	54	0.745	0.778	0.451	0.654	**0.785**
16	sp	164	0.768	0.826	0.536	0.705	**0.843**
17	sp	112	0.584	0.653	0.497	0.634	**0.723**
18	sp	238	0.704	0.803	0.706	0.610	**0.823**
19	sp	450	0.556	0.635	0.441	0.505	**0.682**
20	sp	170	0.511	**0.530**	0.141	0.294	0.479
mean	all	203	0.578	0.645	0.516	0.499	**0.705**

No changes were made to the provided baseline classifier; as such it utilized only character trigrams, a minimum document frequency of 5 (character trigrams that appeared less frequently than five times across the corpus were not included in the vocabulary), no text lower casing and no weighting of the bag of words (only using a count-based approach), a one-vs-rest multi-class strategy. It classified documents as unknown, if and only if, the highest and second highest predictions were less than 0.1 apart.

Results showed that the standard character n-gram model performed generally better than the variable length word n-gram or distorted character n-gram model. However, in the vast majority of runs, the soft voting classifier resulted in a higher score than any of the individual models (see Table 2). According to the mean scores obtained for the three individual models, the word n-gram model showed the worst performance. This is in line with the assumption, that authorship manifests itself in

style and thus a text's syntax and morphology rather than in a text's vocabulary. Word n-grams primarily encode lexical information about a document, whereas standard character n-grams and diacritic characters n-grams rather capture the syntactic and morphological characteristics of the text, which further reveal information about an author's writing style.

SVD implementation reduced our algorithm's computational time from approx. 30 minutes to just 180 seconds. In the final evaluation, we, unfortunately, were unable to deploy SVD due to time constraints, which is why the run time for our PAN 2019 model is 30 minutes and not 180 seconds. The macro F1-score (approx. 70%), however, did not change as a result of dimensionality reduction.

4 Discussion

Results displayed that the soft voting procedure notably outperformed the individual classifiers. Intuitively, one might assume that averaging over classifiers with lower macro F1-scores might yield a worse and not a better performance. However, we did not perform averaging across the F1-scores, but across the predicted probabilities $\hat{y}_i$ obtained for each classifier. This served as an additive factor as all three feature representations were combined into one thorough representation.

One limitation to our approach might have been that we did not apply a weighted voting procedure. This could and shall be addressed in future research. Moreover, it might be interesting to consider a hard instead of a soft voting procedure. We further encourage others to deploy different machine learning classifiers, such as Multinomial Logistic Regression, Multinomial Naive Bayes or Neural Networks. Additionally, further experiments might compute Principal Component Analysis (PCA) instead of Singular Value Decomposition (SVD) to reduce dimensionality and enhance computational efficiency.

5 Conclusion

This paper presented our soft voting ensemble classifier for cross-domain authorship attribution. We combined a standard character n-gram model, a distorted character n-gram model and a word n-gram model to achieve more accurate predictions than the individual models themselves. All three models represented the texts as variable length n-grams, which were weighted by term frequency-inverse document frequency (tf-idf). Our algorithm can be perceived as an enhancement of the PAN 2019 baseline system. One may infer from the results we obtained, that authorship attribution models generally benefit from the inclusion of different representations of text. It is not a single feature representation that will yield accurate classifications, but rather the combination of various document representations that will depict an author's writing style.

References

1. Can, M.: Authorship attribution using principal component analysis and competitive neural networks. Mathematical and Computational Applications **19**(1), 21–36 (2014). https://doi.org/https://doi.org/10.3390/mca19010021
2. Custódio, J.E., Paraboni, I.: EACH-USP ensemble cross-domain authorship attribution. In: Proceedings of the Ninth International Conference of the CLEF Association (2018)
3. Daneshvar, S., Inkpen, D.: Gender Identification in Twitter using N-grams and LSA Notebook for PAN at CLEF 2018. In: Proceedings of the Ninth International Conference of the CLEF Association (2018)
4. Houvardas, J., Stamatatos, E.: N-Gram Feature Selection for Authorship Identification. In: International conference on artificial intelligence: Methodology, systems, and applications. pp. 77–86. Springer, Berlin, Heidelberg (2006). https://doi.org/10.1007/11861461_10
5. Koppel, M., Schlier, J., Argamon, S.: Computational methods in authorship attribution. Journal of the American Society for Information Science and Technology **60**(1), 9–26 (2009). https://doi.org/10.1002/asi.20961
6. Markov, I., Stamatatos, E., Sidorov, G.: Improving cross-topic authorship attribution: The role of pre-processing. In: International Conference on Computational Linguistics and Intelligent Text Processing. pp. 289–302. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-77116-8_21
7. Milli, S., Bamman, D.: Beyond Canonical Texts: A Computational Analysis of Fanfiction. In: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing. pp. 2048–2053 (2016). https://doi.org/10.18653/v1/d16-1218
8. PAN: PAN @ CLEF 2019: Cross-domain Authorship Attribution (2019), https://pan.webis.de/clef19/pan19-web/author-identification.html
9. Pedregosa, F., Varoquaux, G., Gramfort, A., Thirion, B., Michel, V., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Vanderplas, J., Cournapeau, D., Dubourg, V., Passos, A., Brucher, M., Perrot, M., Duchesnay, Ã.: Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research **12**, 2825–2830 (2011)
10. 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)
11. Sapkota, U., Bethard, S., Montes, M., Solorio, T.: Not All Character N-grams Are Created Equal: A Study in Authorship Attribution. In: Proceedings of the 2015 conference of the North American chapter of the association for computational linguistics: Human language technologies. pp. 93–102. Association for Computational Linguistics (ACL) (2015). https://doi.org/10.3115/v1/n15-1010
12. Stamatatos, E.: A Survey of Modern Authorship Attribution Methods. Journal of the American Society for information Science and Technology **60**(3), 538–556 (2009). https://doi.org/https://doi.org/10.1002/asi.21001