Feature Similarity-based Regression Models for Authorship Verification

Notebook for PAN at CLEF 2021

Marina Pinzhakova[1], Tom Yagel[1] and Jakov Rabinovits[1]

[1]*University of Copenhagen, Centre for Language Technology, Department of Nordic Studies and Linguistics, Karen Blixens Vej 8, 2300 Copenhagen, Denmark*

Abstract

This article outlines our strategy in building a machine learning model for the PAN 2021 Authorship Verification Task. While processing every text-pair provided, we used different features based on fundamental principles and intuitions of stylometry to try distinguishing between authors. As our measure of text-pair similarity, we used Cosine distance in vector-based features and absolute difference in scalar features. The similarity scores were then concatenated and used as input to our chosen regression models. We compare the performance of three regression models: Logistic Regression, Support Vector Machines (SVM) and Random Forest Classifier (adapted as a regressor). The Random Forest model achieved the highest AUC of 0.986 on the entire dataset provided by PAN. Finally, we discuss some prospects regarding methods to potentially improve performance of our model and we consider different potential angles for further research.

1. Introduction

This article describes the manner in which we tackle the PAN 2021 Authorship Verification Task. The aim of this task is to build a machine learning model, which will be capable of predicting whether or not two given texts were written by the same author. This task poses more challenges compared with other text classification problems e.g. Author Attribution. While if one seeks to determine if a text was written by one known author or the other, a common strategy would be to train the model to be familiarised with the distinctive writing style of the known author, followed by an attempt to distinguish texts written by unknown authors from that style. However, in Authorship Verification, one must train a model which will be able to differentiate between a subtler set of variances, which either exemplify conscious or unconscious changes in writing style of the same author and possibly greater variances which may indicate styles of different authors. In an attempt to resolve the challenge posed by Authorship Verification, we experiment with various linguistic and linguistically-inspired

CLEF 2021 – Conference and Labs of the Evaluation Forum, September 21–24, 2021, Bucharest, Romania

ptr273@alumni.ku.dk (M. Pinzhakova); tjw403@alumni.ku.dk (T. Yagel); kbj138@alumni.ku.dk (J. Rabinovits)
https://github.com/marinapinzhakova/LP-2-shared-task (M. Pinzhakova);
https://github.com/tomyagel/LP-2-shared-task (T. Yagel); https://github.com/YaShock/LP-2-shared-task (J. Rabinovits)

features that are either intuitive, follow common linguistic practice, or have shown proven success in similar projects. Traditionally speaking, a solution to this problem is used to unveil the identity of otherwise anonymous authors using specific characteristics we call features. These features are derived from textual idiosyncrasies (e.g. grammatical patterns) and stylistic patterns. They can be learned by a model in a way that they are mostly unaffected by the topic or theme of the text, while on the downside, some of them are not linguistically interpretable. We hypothesize that by using an adequate number of features we would be able to train a model capable of distinguishing between authors of different texts to a satisfactory level.

2. Our Approach

Our strategy consists of the following steps: first, we extract vector-based and scalar features from both texts in a text-pair, followed by calculating distances between the texts. The distance measures we apply are Cosine distance on vector-based features and absolute distance on scalar features. The smaller the distance the more similar the texts are presumed to be, thus inferring they are more likely to be written by the same author. We concatenate all the extracted similarities into a matrix, which is used as input to each of our four regression models: Logistic Regression, Support Vector Machines (SVM), and Random Forest. Finally, we evaluate our model using AUC (area under the curve), F1-score, c@1, F_0.5u and Brier scores using the evaluation code provided by PAN (described in further detail under "Evaluation").

2.1. The Data

The dataset provided for this task was compiled by Bischoff et al. [6] and includes texts from fanfiction.net, written in English. The fanfiction.net website contains extensions of fictional storylines (based on existing fiction) written by fans in which an alleged "fandom topic" describes the main subject in a given document. Each record in the dataset consists of two documents which may or may not be written by the same person and the fandom that each document was categorised under. The ground truth specifies the author identifiers for each document and the prediction target indicating if the two documents were written by the same person. The training dataset for the shared task was available in two sizes: a smaller dataset with 52,590 records and a larger dataset with 275,486 records, with each document containing about 21,000 characters and 4,800 tokens.

2.2. Feature Selection

The motivation for our feature selection was largely influenced by frequently used features in previous studies involving linguistic style detection. The selection process was also based on the forensic linguistic method called Writeprint, which is primarily used for identifying different authors online. We also intuitively presumed that other linguistic or linguistically-inspired features might assist us in this task, and we decided to test features originating in various linguistic domains. Such domains include lexicon, syntax and linguistic structure, while other content-based pseudo-linguistic features were also tested. As previously mentioned, we have divided our feature list into vector-based and scalar features as our model treats these two types

in a slightly different way. Some features could be represented in the vector-space. In this way, they serve as a natural metric for comparison: Cosine similarity which measures the angle between vectors, from which we calculate Cosine distance. Nonetheless, other features could only be expressed using a single number (scalar), since they are a result of defined counting methods, ratios and more elaborate mathematical formulae.

2.2.1. Vector-based Features

The following features are expressed in the vector-space and are computed in terms of TF-IDF (Term Frequency – Inverse Document Frequency). We chose to use TfidfVectorizer from the Scikit-learn Python library [16] to calculate the TF-IDF vectors for the text-pairs in all of the following features, except for the punctuation frequency feature:

- **Character n-grams:** values for character n-grams. This is a lexical feature aimed to detect linguistic style based on character sequences. We define a character as any computer-accepted representation of a written symbol occupying a space in a text (including whitespace characters).
- **Word n-grams:** values for word n-grams. This is a lexical feature aimed to detect linguistic style based on word sequences. We define a word as a sequence of characters separated by either a whitespace character or a punctuation mark. We used the NLTK tokenizer's word_tokenize [5] to generate our words.
- **POS-tag (part-of-speech-tag) n-grams:** values for POS-tag n-grams. This is a syntactic feature aimed to detect linguistic style based on POS sequences. POS refers to any syntactic category assigned to a given word. POS tagging is a supervised learning solution that uses features like the previous or next word, first-letter capitalization etc. We used NLTK's pos_tag method to obtain POS tags after performing the word tokenization process.
- **POS-tag chunk n-grams:** values for POS-tag chunk n-grams. This is a syntactic feature generated from partial parsing of syntactic trees assuming the X-bar theory of syntactic category formation, creating a syntactic sub-tree. The tokens we considered are the second-level tokens found on the syntactic tree.
- **Punctuation Frequency:** a syntactic feature aimed to detect linguistic style based on the usage of punctuation marks. We use it to counts all punctuation marks in a given text. In this case, we counted the instances manually without any use of a library counter and/or vectorizer.
- **Stopwords vectorizer:** a syntactic feature aimed to detect linguistic style based on the usage of stopwords. These words traditionally carry little to no meaning and can easily be ignored without compromising the overall meaning of a sentence. We used a pre-defined list from NLTK during our counting step.

Additionally, we used Scikit-learn's CountVectorizer as an alternative way to describe character n-gram counts. In our experiment, all n-gram notions mentioned above refer to bigrams (n=2), apart from POS-tag n-grams where in addition to bigrams we experimented with unigrams (n=1). After extracting the above features, the similarity measure is calculated using Cosine distance, which is 1- Cosine similarity, shown below:

$$\text{similarity} = \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}},$$

and:

$$\text{distance} = 1 - \text{similarity}$$

While A and B are vector-based features from each text in a text-pair, n is the number of text-pairs and θ is the angle between the vectors expressing the features in each text. Note that unlike scalar features (detailed in the next subsection), vector-based features had to be fitted first before being transformed and ready to be used further in our experiment.

2.2.2. Scalar Features

The following features were extracted by performing calculations resulting in a single number (scalar):
- **Vocabulary Richness:** a lexical feature based on a standard richness measure defined by a Type-Token ratio.
- **Average Sentence Lengths:** a lexical feature based on the average number of words per sentence in a given text.
- **Flesch Reading Ease Score:** a multivariate feature combining the count of syllables, words and sentences. This score is used for measuring text readability. It is meant to indicate how difficult it is to read a given text. This score is calculated using the following formula:

$$\text{Readability Ease} = 206.835 - (1.015 \cdot \text{ASL}) - (84.6 \cdot \text{ASW})$$

Where ASL stands for the average sentence length and ASW stands for the average number of syllables per each word.

We computed absolute differences of scalar feature values for each pair of texts. These differences, together with the Cosine distances of the vector-based features, were then concatenated into matrices which serve as input to our regression models.

2.2.3. Feature Importance

During our experimentation process, we have tested many types of features, including additional ones that have not been presented in the article so far. These features include Yule's Richness Score, word lengths, emoticons and foul language. However, only the features that made the most impact were chosen moving forward. We used the feature importance parameter provided by the Random Forest classifier in an attempt to discover which features matter the most. In **Table 1** we show the detailed normalised feature importance scores used by the Random Forest classifier (adapted as a regressor) for each feature we considered using over the entire dataset.

Table 1
Examined features (both included and excluded in our experiment) and their corresponding importance scores determined by the Random Forest Classifier

Feature name	Feature Importance Score (Normalised)
Word bigrams (TF-IDF)	0.315
Punctuation frequency vectorizer	0.102
POS bigrams	0.096
Stopword vectorizer	0.096
POS-tag chunk bigrams	0.065
Character bigrams (Count)	0.061
Vocabulary Richness	0.059
Character bigrams (TF-IDF)	0.046
Word bigrams (TF-IDF)	0.315
Punctuation frequency (vectorizer)	0.102
POS-tag bigrams	0.096
Stopword unigrams	0.096
POS-tag chunk bigrams	0.065
Character trigrams (Count)	0.061
Vocabulary richness	0.059
Character trigrams (TF-IDF)	0.046
POS-tag unigrams	0.036
Punctuation frequency (scalar)	0.019
Stopword frequency (version 1)	0.018
Average sentence length	0.017
Flesch reading ease score	0.014
Number of emoticons	0.012
Average word length	0.010
Curse word frequency	0.010
Consecutive capitalization	0.009
Stopword frequency (version 2)	0.008
Words with consecutive characters	0.007

The scalar features which were meant to extract frequencies of punctuation marks and stopwords (named 'Punctuation frequency' and 'Stopword frequency' accordingly) were essentially capturing the same information as their vector-based equivalents. However, they showed significantly lower importance scores, making it easier for us to decide to discard them and move ahead with some other scalar features instead, next in the level of importance.

2.3. Regression Models

We approached the Authorship Verification Task as a regression problem. The model's prediction output is a value between 0 and 1, where 1 is a complete confidence that two texts are written by the same author. When calculating scores, we use binarization with a cut-off of 0.5. The training dataset (which was the entire dataset provided by PAN) was split into training-set (80%) and test-set (validation - 20%). The best performing model selected was the one producing the lowest value of mean square error (calculated and averaged) under 5-fold cross-validation. We

experimented with the following supervised learning models and hyperparameters:
- **Logistic Regression** with C (inverse of regularization strength) = [0.001, 0.01, 0.1, 1, 10].
- **Support Vector Machines** with C (inverse of regularization strength = [0.001, 0.01, 0.1, 1, 10].
- **Random Forest** with nr_estimators (number of trees) = [50, 100, 150, 200] and max_depth (number of nodes) = [5, 10, 15, 20].
Note that the feature matrix was normalised prior to being processed by the Support Vector Machines model. Random Forest (200 trees, 20 nodes) was selected as the best model and was subsequently run in the TIRA system. The model was then trained on the entire dataset. All the models used are implemented by the Python Scikit-learn library.

2.3.1. Baseline

The baseline, which was provided by PAN, is a TF-IDF-weighted character n-grams, where the distance measure is expressed in Cosine distance. The baseline then uses a compression method to calculate the cross-entropy. The documents (texts) provided in the baseline are represented with a bag-of-character-n grams model that is TF-IDF-weighted. The Cosine distance measure between each document-pair in the calibration dataset is then calculated. Following an optimisation process, the previously-calculated similarities function as pseudo-probabilities indicating the likelihood that two documents in a given pair were written by the same author.

2.4. Evaluation

In order to examine the performance of our model, we used the evaluation platform provided by PAN, which includes the following parameters: AUC: the conventional area-under-the-curve score, as implemented in Scikit-learn. F1-score: the well-known performance measure (not taking into account non-answers), as implemented in Scikit-learn. c@1: a variant of the conventional F1-score, which rewards systems that leave difficult problems unanswered (i.e. scores of exactly 0.5), introduced by Peñas and Rodrigo (2011). F_0.5u: a newly proposed measure that puts more emphasis on deciding same-author cases correctly (Bevendorff et al. 2019). Brier: the complement of the well-known Brier score, for evaluating the quality of (binary) probabilistic classifiers, as implemented in Scikit-learn.

3. Results

We present our results in two parts. The first part (shown in **Table 2**) is the outcome of our own internal evaluation during best model selection using train/test split. The results are shown for the Random Forest classifier (adapted as a regressor), which was the best performing model.

The second part (shown in **Table 3**) is the outcome of the evaluation code provided by PAN for baseline and our Random Forest model. Both models were trained over the entire training dataset and then evaluated upon it. The baseline model was trained with the vocabulary size set to 2,000 and the n-value of n-grams to 2. At a later stage we will be able to provide full results for the entire dataset.

Table 2
Results using our own metrics and evaluation techniques.

	Training Score	Test (Validation) Score
MSE	0.015	0.013
AUC	0.986	0.893
Accuracy	0.985	0.816
F1	0.986	0.820

Table 3
Results using the evaluation code provided by PAN.

	Baseline	Random Forest
F1	0.794	0.986
AUC	0.825	0.986
c@1	0.771	0.985
F_0.5	0.726	0.994
Overall	0.779	0.988

Table 4
Results using metrics and evaluation techniques over the official test set provided by PAN.

	Test Score
AUC	0.8129
c@1	0.8129
F1	0.8094
F0.5u	0.8186
Brier	0.8129
Overall	0.8133

4. Discussion and Conclusion

In this article, we described our strategy in building a machine learning model for the PAN 2021 Authorship Verification Task, where the aim was detecting whether or not two texts were written by the same author. Our strategy included the extraction of stylometric features from a given text-pair and calculating similarity measures between each of the texts. We used Cosine distance in vector-based features and absolute difference in scalar features. The similarity measures were then concatenated and used as input to each of our pre-selected regression models. The model was both internally evaluated under our own pre-set metrics and externally assessed by the PAN evaluation mechanism. Unlike the approach required to tackle Authorship Attribution problems, where a list of authors is known in advance, we offer a more generalised model applicable to any text-pairs without any prior knowledge about specific authors. The fact that our model performed well in an open-set verification setting, where the authors and topics are novel, reaffirms our confidence in the model we built. It also turns out that specific model selection did not have as much impact as we suspected to begin with, as performance

was mostly influenced by the features we selected. For example, word and POS-tag frequencies were by far the most dominant ones. This can be explained intuitively as words and POS-tags are among the foundations of any written work (as they are in other linguistic paradigms), and we can easily see them as good "profilers" for writing style. For future research we could conduct a more thorough investigation regarding feature importance. We might then see new features added and existing ones removed. Furthermore, we believe it would be interesting to experiment with different values for the n's in the n-gram sequences we generated. Finally, we imagine using different syntactic trees' formation approaches than X-bar, namely dependency grammar and context-free grammar, such as Chomsky's normal form. Either of these has the potential to produce compelling results.

Acknowledgements

We thank the PAN2021 organisers for arranging the shared task and helping us through the registration and submission process. We also thank our lecturer, Manex Agirrezabal Zabaleta for his encouragement to partaking in this shared task. Our work was supported by the DISF study foundation in memory of Josef and Regine Nachemsohn.

References

A. Online Resources

The code is available via

- GitHub
- Overleaf template.

B. Other Resources

[1] Abbasi, A., Chen, H.c.: Writeprints: A stylometric approach to identity-level identification and similarity detection in cyberspace. ACM Transactions on Information Systems 26, 1–29 (01 2008).
https://doi.org/10.1145/1344411.1344413

[2] Baayen, H., H. van Halteren, F. Tweedie (1996). Outside the cave of shadows: Using syntactic annotation to enhance authorship attribution, Literary and Linguistic Computing, 11, 1996.

[3] Bevendorff, J., Stein, B., Hagen, M., Potthast, M.: Generalizing unmasking for short texts. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers). pp. 654–659. Association for Computational Linguistics, Minneapolis, Minnesota (Jun 2019).
https://doi.org/10.18653/v1/N19-1068,
https://www.aclweb.org/anthology/N19-1068

[4] Bevendorff, J., Chulvi, B., Liz De La Peña Sarracén, G., Kestemont, M., Manjavacas, E., Markov, I., Mayerl, M., Potthast, M., Rangel, F., Rosso, P., Stamatatos, E., Stein, B., Wiegmann, M., Wolska, M., and Zangerle, E., 12th International Conference of the CLEF Association (CLEF 2021), Candan, K.S., Ionescu, b., Goeuriot, L., Larsen, B., Müller, H., Joly, A., Maistro, M., Piroi, F., Faggioli, G., and Ferro, N., Springer Publishing, Overview of PAN 2021: Authorship Verification,Profiling Hate Speech Spreaders on Twitter,and Style Change Detection, Bucharest, Romania, 2021.

[5] Bird, S., Klein, E., Loper, E.: Natural language processing with Python: analyzing text with the natural language toolkit. "O'Reilly Media, Inc." (2009).

[6] Bischoff, S., Deckers, N., Schliebs, M., Thies, B., Hagen, M., Stamatatos, E., Stein, B., Potthast, M.: The Importance of Suppressing Domain Style in Authorship Analysis. CoRR abs/2005.14714 (May 2020),
https://arxiv.org/abs/2005.14714

[7] Boenninghoff, B., Rupp, J., Nickel, R.M., Kolossa, D., Deep Bayes Factor Scoring for Authorship Verification, Ruhr University Bochum, Germany and Bucknell University, Lewisburg, PA, USA (2020) as part of CLEF 2020, 22-25 September 2020, Thessaloniki, Greece.

[8] De Vel, O., M. Corney, A. Anderson and G. Mohay (2002), E-mail Authorship Attribution for Computer Forensics, in Applications of Data Mining in Computer Security, Barbará, D. and Jajodia, S. (eds.), Kluwer.

[9] Ehrhardt, S.: Authorship attribution analysis. In: Visconti, J. (ed.) Handbook of Communication in the Legal Sphere. pp. 169–200. de Gruyter, Berlin/Boston (2018).

[10] Hirst, G., Feiguina, O.: Bigrams of syntactic labels for authorship discrimination of short texts. Literary and Linguistic Computing 22(4), 405–417 (2007).

[11] Kestemont, M., Markov, I., Stamatatos, E., Manjavacas, E., Bevendorff, J., Potthast, M., Stein, B.: Overview of the Authorship Verification Task at PAN 2021. In: Faggioli, G., Ferro, N., Joly, A., Maistro, M., and Piroi, F., (eds.) CLEF 2021 Labs and Workshops, Notebook Papers. CEUR-WS.org (2021).

[12] Koppel, M., Schler, J., Authorship Verification as a One-class Classification Problem, Dept. of Computer Science, Bar-Ilan University, Ramat-Gan, Israel (2004).

[13] Koppel M., Schler J., Argamon S., Computational methods in authorship attribution. Journal of the American Society for Information Science and Technology. 2009;60(1):9-26. doi:10.1002/asi.20961

[14] Luyckx, K., Daelemans, W.: Shallow text analysis and machine learning for authorship attribution. LOT Occasional Series 4, 149–160 (2005).

[15] Mosteller, F. and Wallace, D. L. (1964). Inference and Disputed Authorship: The Federalist. Reading, Mass.: Addison Wesley.

[16] 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).

[17] Peñas, A., Alvaro Rodrigo: A simple measure to assess non-response. In: ACL. ´ pp. 1415–1424 (2011),
http://www.aclweb.org/anthology/P11-1142

[18] Potthast, M., Gollub, T., Wiegmann, M., and Stein, B.,Information Retrieval Evaluation in a Changing World, doi: 10.1007/978-3-030-22948-1_5, edited by Ferro, N., and Peters, C., isbn = 978-3-030-22948-1, Springer Publishing, The Information Retrieval Series, TIRA Integrated Research Architecture, September 2019.

[19] Sidorov, G., Velasquez, F., Stamatatos, E., Gelbukh, A., Chanona-Hern´andez, L.: Syntactic n-grams as machine learning features for natural language processing. Expert Systems with Applications 41(3), 853 – 860 (2014).
https://doi.org/https://doi.org/10.1016/j.eswa.2013.08.015,
http://www.sciencedirect.com/science/article/pii/S0957417413006271,
methods and Applications of Artificial and Computational Intelligence.

[20] Stamatatos E. A survey of modern authorship attribution methods. Journal of the American Society for Information Science and Technology. 2009;60(3):538-556.
doi:10.1002/asi.21001

[21] Tweedie, F. J. and R. H. Baayen (1998). How Variable May a Constant Be? Measures of Lexical Richness in Perspective, Computers and the Humanities, 32 (1998), 323-352.

[22] Weerasinghe, J., Greenstadt, R., Feature Vector Difference based Neural Network and Logistic Regression Models for Authorship Verification, New York University, as part of CLEF 2020, 22-25 September 2020, Thessaloniki, Greece.

[23] Yule, G.U. (1938). On Sentence Length as a Statistical Characteristic of Style in Prose with Application to Two Cases of Disputed Authorship, Biometrika, 30, 363-390.