Feature Vector Difference based Authorship Verification for Open-World Settings

Notebook for PAN at CLEF 2021

Janith Weerasinghe[1], Rhia Singh[2] and Rachel Greenstadt[1]

[1]New York University, 6 MetroTech Center, Brooklyn, NY 11201, United States of America
[2]Macaulay Honors College (Hunter CUNY), 695 Park Avenue, New York, NY 10065, United States of America

Abstract

This paper describes the approach we took to create a machine learning model for the PAN 2021 Authorship Verification Task. The goal of this task is to predict if a given pair of documents are written by the same author. For each document pair, we extracted stylometric features from the documents and used the absolute difference between the feature vectors as input to our classifier. Our new model is similar to out last year's model with minor improvements to the feature set and the classifier. We trained two models on the two small and large datasets which achieved AUCs of 0.967 and 0.972 in the final evaluations.

Keywords

Authorship Verification, Stylometry, Machine Learning, Natural Language Processing

1. Introduction

This paper presents our approach for the Authorship Verification Shared Task [1] at PAN 2021[2]. The objective of this task was to create a model that would be able to predict if two given documents were written by the same person. This year's shared task used the same training data as last year, but is more challenging because the "test set [is] made entirely of unseen authors and topics."[1] This requires our model to be both topic agnostic and work robustly in an open-world setting. Our new model follows our approach[3] from the PAN2020 authorship verification task [4] with improvements made to address these new challenges.

The dataset provided for this task was compiled by Bischoff et al. [5] and contains English documents from fanfiction.net. 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 categorized 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 on average about $21,000$ characters and $4,800$ tokens.

CLEF 2021 – Conference and Labs of the Evaluation Forum, September 21–24, 2021, Bucharest, Romania
janith@nyu.edu (J. Weerasinghe); rhia.singh@macaulay.cuny.edu (R. Singh); greenstadt@nyu.edu (R. Greenstadt)

[1]https://pan.webis.de/clef21/pan21-web/author-identification.html

2. Approach

The goal of the PAN 2021 authorship verification shared task [2] was to predict if two given documents (D_i and D_j) were written by the same person. We modeled this as a binary classification problem, in which the input to our classifier is a feature vector encoding the two documents (X) and the target variable (Y), indicating whether or not the two documents were written by the same author.

We trained two models (trained on the smaller and larger datasets) using an identical approach for both datasets. Our approach was implemented on Python primarily using NLTK [6] and Scikit Learn [7] libraries, and the source code is available at: https://github.com/janithnw/pan2021_authorship_verification.

2.1. Preparing the datasets

We divided the provided small and large datasets into training and testing sets, with the training sets roughly containing 75% of the total records. Since the model needs to work in an open-world setting, we needed to ensure that the authors included in the training set did not appear in the test set. To do this split, we started by iterating through all the authors in the dataset. Then, we randomly decided for each author if they would be placed in the training set (with a 75% chance) or the testing set (with a 25% chance). Then for each selected author, we also find all the other authors that appear with them in the dataset and include those authors and their associated records into the same partition as the original author. Note that we do this search in a recursive manner so that once a single author is included in a partition, all the authors that they appear with, and all the other authors that the newly selected authors appear with, are all included in the same training or test dataset. However, we did not take into account the topics (or fandoms) when doing these splits. Therefore it is possible that both the training and testing sets contain fan-fictions written about the same fandom.

2.2. Preprocessing

We ran each document in the dataset through a series of pre-processing steps before feature extraction. The outputs of the preprocessing steps are stored together with the document, which is passed to the feature extraction step in our pipeline. We will use the following sentence as a running example in this section:

> "The Soviets had already been merciless, ruthless as the next army."

Tokenizer: We used the NLTK's `casual_tokenize` method, which uses their `TweetTokenizer` to tokenize the documents. Our initial observations found this method to perform better at handling punctuation marks and words than the default Treebank Word Tokenizer. The tokenized version of the document is stored to be used in the next pre-processing steps and to be used in feature extraction steps.

Part-of-Speech (POS) Tagging: We used NLTK's Perceptron Tagger to perform the parts of speech tagging. The POS tags are stored together with the document, which are used in the next preprocessing steps and in feature extraction. The following would be the output of our POS-tagger for the example sentence above:

```
[('The', 'DT'), ('Soviets', 'NNPS'), ('had', 'VBD'),
 ('already', 'RB'), ('been', 'VBN'), ('merciless', 'RB'),
 (',', ','), ('ruthless', 'NN'), ('as', 'IN'), ('the', 'DT'),
 ('next', 'JJ'), ('army', 'NN'), ('.', '.')]
```

Generating a Partial Parse Tree (POS Tag Chunking): We trained a Maxent (Maximum Entropy) classifier using the CoNLL 2000 corpus[8] to do POS tag chunking following the example provided by Bird*et al.* [9] in their NLTK book (Chapter 07). The following would be the output of our parser for the example sentence above:

```
(S
  (NP The/DT Soviets/NNPS)
  (VP had/VBD already/RB been/VBN)
  (NP merciless/NN)
  ,/,
  (NP ruthless/NN)
  (PP as/IN)
  (NP the/DT next/JJ army/NN))
```

2.3. Features

This section lists the features that we extracted from the preprocessed data. Most of these features are used in our previous work [3] and are commonly used in previous stylometry work [10]. We used some features that are described in Writeprints feature set [11]. We also believed that the syntactic structure of sentences would provide valuable signals to the classifier. Following prior work [12, 13], we included POS-Tag n-grams and partial parses (or POS-Tag chunks) as part of our feature set. The use of parse trees to extract stylometric features, called syntactic dependency-based n-grams of POS tags, was introduced by Sidorov et al. [14]. We used a slightly different approach to encode parse tree features (described below), capturing how different noun and verb phrases are constructed.

Several of our features described below are computed in terms of TF-IDF values. We used SKLearn's `TFIDFVectorizer` to compute the TF-IDF vectors for the documents. We set the `min_df` parameter to be 0.1 to ignore tokens that have a document frequency less than 10%. Features that are new or changed in this year's model are denoted with an asterisk (*).

- **Character n-grams***: TF-IDF values for character n-grams, where $1 \geq n \geq 3$. In our last year's model, we included up to character-6-grams. We believed this resulted in our model being slightly affected by topic similarities[15]. To avoid this bias, our current model only includes character tri-grams.
- **POS-Tag n-grams**: TF-IDF value of POS-Tag tri-grams.
- **Special Characters**: TF-IDF values for 31 pre-defined special characters.
- **Frequency of Function Words***: Frequencies of 851 common English words[2].
- **Average number of characters per word**: The average number of characters per token.
- **Distribution of word-lengths (1-10)**: The fraction of tokens of length l, where $1 \leq l \leq 10$

[2]Downloaded from https://countwordsfree.com/stopwords

- **Vocabulary Richness***: In this year's model, we included several measures of vocabulary richness. The first is the ratio of hapax-legomena and dis-legomena, which was included in last year's model. Here, hapax-legomena is the number of words that only occur once in the document and dis-legomena is the number of words that occur twice. In addition, we included the following measures: Type-token ratio, Guiraud's R[16], Herdan's C[17, 18], Dugast's k and U[19], Maas' a^2[20], Tuldava's LN[21], Brunet's W[22], Carroll's CTTR[23], Summer's S, Sichel's S[24], Michéa's M[25], Honoré's H[26], Herdan's V_m[27], entropy, Yule's K[28], and Simpson's D[29]. We used the implementation of these algorithms in the Python `textcomplexity` package[3].
- **POS-Tag Chunks**: TF-IDF values for Tri-grams of POS-Tag chunks. Here, we consider the tokens at the second level of our parse tree. For example, for the sentence above, the input to our vectorizer would be `['NP', 'VP', 'NP', ',', 'NP', 'IN', 'NP', '.']`.
- **POS chunk construction**: TF-IDF values of each noun phrase, verb phrase, and prepositional phrase expansion. For the sentence above, these expansions are `['NP[DT NNPS]', 'VP[VBD RB VBN]', 'NP[NN]', 'NP[NN]', 'NP[DT JJ NNP]']`
- **Stop-word and POS tag hybrid tri-grams***: To capture stylistic information about word order while also preventing topic related biases, we replaced all the words other than the function words with their part-of-speech tag and computed the TF-IDF values of the tri-grams from this modified text. Similar methods of *text distortion* have been used successfully in previous studies[30, 31].
- **Part-of-Speech tag ratios*** Following the work of Castro-Castro *et al.* [32] who computed the ratio of nouns and adjectives, we calculated the proportion of all parts of speech tags in the Penn Treebank POS Tag collection in an attempt to better capture the syntactic composition of the text.
- **Unique spellings***: The fraction of words that are present in the document that belong to each of the following dictionaries: commonly misspelled English words[4], common typos when communicating online [5], common errors with determiners [6], British spelling of words [7], and popular online abbreviations [8][9].

We fit our feature extractors on the training sets. We also standardize features by removing the mean and scaling to unit variance. Then, we take the absolute vector difference between the feature vectors corresponding to each document pair. We then apply a secondary scaling step to ensure that the vector differences are standardized as well. This step was necessary for the stochastic gradient descent algorithm that we used to train our logistic regression classifier. More formally, given documents D_i and D_j, we represent their scaled feature vectors as X_i and X_j. Then we compute the vector difference as $X = |X_i - X_j|$. Then the input to our classifier will be the scaled version of X.

2.4. Classifier

We computed the features for each document pair in the two datasets (smaller and larger) as described in the previous section. Our previous experience showed that Logistic Regression classifier worked the best with our approach. Since the complete feature matrix cannot be stored in-memory, we used a Stochastic Gradient Descent training algorithm with a logarithmic loss function, which results in a logistic regression classifier. We used SKLearn's `SGDClassifier` implementation. We found the best value for the *alpha* parameter using `RandomizedSearchCV` and running the search on a sample of training records. We ran the `SGDClassifier` for 50 iterations.

[3] https://github.com/tsproisl/textcomplexity
[4] https://www.mentalfloss.com/article/629813/100-commonly-misspelled-words-english
[5] https://www.lexico.com/grammar/common-misspellings
[6] https://www.ef.edu/english-resources/english-grammar/determiners/
[7] https://www.lexico.com/grammar/british-and-spelling
[8] https://preply.com/en/blog/2020/05/07/the-most-used-internet-abbreviations-for-texting-and-tweeting
[9] https://englishstudyhere.com/abbreviations-contractions/50-common-internet-abbreviations/

3. Results

Table 1 shows the results of our two models under different test datasets and settings. The models are evaluated on 5 measures: area under the ROC curve (AUC), F1-score, c@1 (a variant of the F1-score, which rewards systems that leave difficult problems unanswered [33]), F_0.5u (a measure that puts more emphasis on deciding same-author cases correctly [34]), and the complement of the Brier score [35]. We submitted our smaller model during the early submission phase. These results were obtained before we incorporated the new vocabulary richness measures, stop-word and POS tag hybrid tri-grams features, POS tag ratios, and unique spellings to our feature set. Table 1 shows the results of our two models under different settings. Once the final models were trained, we deployed these models to the TIRA evaluation system [36] provided by the PAN 2021 organizers where the models were evaluated on an unseen dataset.

Table 1
Results from our local evaluations, early submissions, and the final evaluations

Description	AUC	C@1	F0.5U	F1-Score	Brier
Small dataset, local test set	0.965	0.903	0.928	0.903	0.925
Small dataset, early submission	0.955	0.890	0.894	0.889	0.919
Small dataset, Final evaluation	0.967	0.910	0.907	0.927	0.929
Large dataset, local test set	0.967	0.909	0.918	0.915	0.928
Large dataset, Final evaluation	0.972	0.917	0.916	0.926	0.934

4. Discussion and Conclusion

In this paper we presented the approach for an authorship verification model that works robustly in an open-world setting under varying topics. Our approach is an improvement over our earlier model which was submitted to PAN 2020 Authorship Verification task. Most of the improvements are made by incorporating new features.

We would also like to discuss other ideas that we attempted to include but were not successful. We attempted to split each document into a few smaller documents and train a model on a larger number of smaller documents. We hoped by doing so we will be able to simulate having multiple documents per author and therefore be able to do multiple comparisons across the two authors. We expected the aggregated results of multiple comparisons would result in better performance or serve as another measure for classifier confidence which we could then use to leave out low confidence predictions. Our current attempts to use this strategy did not result in better performance values. We also attempted to encode features extracted from dependency parses. The performance gain after incorporating these features were very minimal. We ended up not including the dependency parse features due to the significant computing power required to do dependency parsing.

Our work shows that, by selecting features that are less likely to encode topic information, it is possible to train a topic agnostic authorship verification model that works well in open-world settings.

5. Acknowledgements

We thank PAN2021 organizers for organizing the shared task and helping us through the submission process. We also thank the reviewers for their helpful comments and feedback. Our work was supported by the National Science Foundation under grant 1931005 and the McNulty Foundation.

References

[1] M. Kestemont, I. Markov, E. Stamatatos, E. Manjavacas, J. Bevendorff, M. Potthast, B. Stein, Overview of the Authorship Verification Task at PAN 2021, in: CLEF 2021 Labs and Workshops, Notebook Papers, CEUR-WS.org, 2021.

[2] J. Bevendorff, B. Chulvi, G. L. D. L. P. Sarracén, M. Kestemont, E. Manjavacas, I. Markov, M. Mayerl, M. Potthast, F. Rangel, P. Rosso, E. Stamatatos, B. Stein, M. Wiegmann, M. Wolska, , E. Zangerle, Overview of PAN 2021: Authorship Verification,Profiling Hate Speech Spreaders on Twitter,and Style Change Detection, in: 12th International Conference of the CLEF Association (CLEF 2021), Springer, 2021.

[3] J. Weerasinghe, R. Greenstadt, Feature Vector Difference based Neural Network and Logistic Regression Models for Authorship Verification—Notebook for PAN at CLEF 2020, in: L. Cappellato, C. Eickhoff, N. Ferro, A. Névéol (Eds.), CLEF 2020 Labs and Workshops, Notebook Papers, CEUR-WS.org, 2020. URL: http://ceur-ws.org/Vol-2696/.

[4] M. Kestemont, E. Manjavacas, I. Markov, J. Bevendorff, M. Wiegmann, E. Stamatatos, M. Potthast, B. Stein, Overview of the Cross-Domain Authorship Verification Task at PAN 2020, in: L. Cappellato, C. Eickhoff, N. Ferro, A. Névéol (Eds.), CLEF 2020 Labs and Workshops, Notebook Papers, CEUR-WS.org, 2020.

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

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

[7] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, Scikit-learn: Machine learning in Python, Journal of Machine Learning Research 12 (2011) 2825–2830.

[8] E. F. Tjong Kim Sang, S. Buchholz, Introduction to the conll-2000 shared task: Chunking, in: C. Cardie, W. Daelemans, C. Nedellec, E. Tjong Kim Sang (Eds.), Proceedings of CoNLL-2000 and LLL-2000, Lisbon, Portugal, 2000, pp. 127–132.

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

[10] E. Stamatatos, A survey of modern authorship attribution methods, Journal of the American Society for information Science and Technology 60 (2009) 538–556.

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

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

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

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

[15] M. Kestemont, E. Manjavacas, I. Markov, J. Bevendorff, M. Wiegmann, E. Stamatatos, M. Potthast, B. Stein, Overview of the Cross-Domain Authorship Verification Task at PAN 2020, in: L. Cappellato, C. Eickhoff, N. Ferro, A. Névéol (Eds.), CLEF 2020 Labs and Workshops, Notebook Papers,

CEUR-WS.org, 2020. URL: http://ceur-ws.org/Vol-2696/.
[16] P. Guiraud, Les caractères statistiques du vocabulaire: essai de méthodologie, Presses universitaires de France, 1954.
[17] G. Herdan, Type-token mathematics, volume 4, Mouton, 1960.
[18] G. Herdan, Quantitative linguistics, london: Buttersworths, See Also (1964).
[19] D. Dugast, Vocabulaire et stylistique, volume 8, Slatkine, 1979.
[20] H.-D. Mass, Über den zusammenhang zwischen wortschatzumfang und länge eines textes [relationship between vocabulary and text length, Zeitschrift für Literaturwissenschaft und Linguistik 2 (1972) 73.
[21] J. Tuldava, Quantitative relations between the size of the text and lexical richness, Journal of Linguistic Calculus (1977) 28–35.
[22] É. Brunet, Le vocabulaire de Jean Giraudoux, structure et évolution, volume 1, Slatkine, 1978.
[23] J. B. Carroll, Language and thought, Reading Improvement 2 (1964) 80.
[24] H. S. Sichel, On a distribution law for word frequencies, Journal of the American Statistical Association 70 (1975) 542–547.
[25] R. Michéa, Répétition et variété dans l'emploi des mots, Bulletin de la Société de Linguistique de Paris (1969) 1–24.
[26] A. Honoré, Some simple measures of richness of vocabulary, Association for literary and linguistic computing bulletin 7 (1979) 172–177.
[27] G. Herdan, A new derivation and interpretation of yule's 'characteristic'k, Zeitschrift für angewandte Mathematik und Physik ZAMP 6 (1955) 332–339.
[28] C. U. Yule, The statistical study of literary vocabulary, Cambridge University Press, 2014.
[29] E. H. Simpson, Measurement of diversity, nature 163 (1949) 688–688.
[30] S. Bergsma, M. Post, D. Yarowsky, Stylometric analysis of scientific articles, in: Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2012, pp. 327–337.
[31] E. Stamatatos, 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, 2017, pp. 1138–1149.
[32] D. Castro-Castro, C. Rodríguez-Losada, R. Muñoz, Mixed Style Feature Representation and B0-maximal Clustering for Style Change Detection—Notebook for PAN at CLEF 2020, in: L. Cappellato, C. Eickhoff, N. Ferro, A. Névéol (Eds.), CLEF 2020 Labs and Workshops, Notebook Papers, CEUR-WS.org, 2020. URL: http://ceur-ws.org/Vol-2696/.
[33] A. Peñas, Álvaro Rodrigo, A simple measure to assess non-response, in: ACL, 2011, pp. 1415–1424. URL: http://www.aclweb.org/anthology/P11-1142.
[34] J. Bevendorff, B. Stein, M. Hagen, M. Potthast, 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), Association for Computational Linguistics, Minneapolis, Minnesota, 2019, pp. 654–659. URL: https://www.aclweb.org/anthology/N19-1068. doi:10.18653/v1/N19-1068.
[35] G. W. Brier, Verification of forecasts expressed in terms of probability, Monthly weather review 78 (1950) 1–3.
[36] M. Potthast, T. Gollub, M. Wiegmann, B. Stein, TIRA Integrated Research Architecture, in: N. Ferro, C. Peters (Eds.), Information Retrieval Evaluation in a Changing World, The Information Retrieval Series, Springer, Berlin Heidelberg New York, 2019. doi:10.1007/978-3-030-22948-1_5.