Authorship Verification

Nilesh Solanki
solank01@ads.uni-passau.de
University of Passau
Passau, Germany

Rohan Sapate
sapate01@ads.uni-passau.de
University of Passau
Passau, Germany

Smruti Ranjan Mohapatra
mohapa01@ads.uni-passau.de
University of Passau
Passau, Germany

ABSTRACT

Authorship verification is the task to determine the authenticity of any document based on the other work of the same author. This task used to be done manually by the experts of this field, but there has been growing researches in the field of Authorship verification problem in the recent years using machine learning to make it more reliable and relatively faster. In this project, we are trying to accomplish PAN 2020 challenge task of authorship verification. In our experiment, we applied a dissimilarity based algorithm on every pair of text to calculate distance between documents and later on applied automatic thresholding to distinguish between the authors.

KEYWORDS

Pan 2020, Text Mining Project, Dissimilarity, Part of speech Tagging, Character n-gram, Word n-gram

ACM Reference Format:
Nilesh Solanki, Rohan Sapate, and Smruti Ranjan Mohapatra. 2020. Authorship Verification. In *Proceedings of Text Mining Lab.* ACM, New York, NY, USA, 5 pages.

1 INTRODUCTION

Authorship analysis is the process of examining the characteristics of two texts in order to draw conclusions if they are written by the same author or not. It is a growing problem in this digitalized world and there has been an extensive study in this area. There are three subfields of authorship analysis: Authorship Profiling, Authorship Identification and Authorship Verification [5]. In the cybercrime world, the culprits are unidentified and thus the task of the forensic experts is to find their age, gender, nationality and other personal details (-in some cases demographics) from the texts they have written which is called authorship profiling. There are a lot of unauthored documents which a lot of people are falsely claiming to be their own. We can solve this problem and find the most likely author of the unknown document when provided with some existing texts from respective authors, which is called authorship identification, which is the second scenario. And the third scenario, and on which we will be working in this project, is called authorship verification. The task of authorship verification is to find out if the given documents are written by the same author. Our emphasis is to discover more details, problems and various approaches required to address authorship verification. Every author has its own writing style, so we can distinguish a particular author from other authors based on the writing style [7]. In this project, we will work on understanding different methodologies used for this task like Common N-Gram profiles of text documents and apply to detect whether the document is written by the author.

Authorship Verification is one of the tasks from PAN 2020. In this task, in the given data, there are pairs of texts given and the objective of the task is to determine whether or not both the documents in each pair are written by the same author. Calculating the dissimilarity scores of the two documents and then automatic assignment of the threshold values, of normalized similarity scores play a vital role in our pursuit to verify the authorship in this task. In this project, we will try to apply the knowledge of text processing, tokenization, segmentation, frequency distribution, conditional frequency distribution, n-gram and so on.

2 PROBLEM STATEMENT

In our experiment we are solving the task of authorship verification where the task is to decide whether two texts of similar length have been written by the same author or not [1]. This is a PAN challenge where there are three problems which are to be studied for three different years 2020, 2021 and 2022 which are respectively closed-set verification problem, open-set verification problem and a surprise task. In our project, we only focus on the closed set verification for this year where a given dataset consists of a large training data set and test dataset which is a subset of training data. However they have provided us only training set, so we have divided that dataset into 90:10 ratio respectively as training and test dataset. The datasets consist of known authors who have written about a set of topics.

3 RELATED WORK

The earlier work done in this domain consists of common n-Gram dissimilarity which was based on the differences in the frequencies of n-grams of tokens (characters, words) that are most common in the considered documents [6]. Several other researches carried out comprise of the "unmasking method" for authorship verification is successful for novel-length texts where different features are taken into consideration from a feature set that comprises of syntactic structure; part of speech n-grams or syntactic and orthographic idiosyncrasies [8]. This method constructs an SVM classifier to identify unknown text from a set of known documents which are all written by a single author [8]. The many candidates problem as mentioned in the paper [9] refers to determining the author of any anonymous document from a provided large set of candidate authors, also known as open-set identification problem. The solution to the many candidate problems is the impostor method which lowers the verification problem by addressing some important issues. The first one being how the impostor set is chosen and second one, to take into consideration the number of impostors to use [9]. The normalised similarity threshold values, as taken in the

previous research papers of authorship verification, were assigned manually. In our project, we have tried to experiment by taking the threshold values of normalised similarity score automatically rather than assigning it manually. Additionally, we have experimented on different features which include Part of speech tag count and various Part of speech tags. We have tried to check which of the considered features provide the best result.

4 DATASET STATISTICS

We acquired the dataset from PAN 2020 organisers which comprises of two files, one for training data of 52,601 records which consists of pair of texts, unique ids and their fandom labels for each row and other for ground truth data of equal number of records as training data which contains boolean flag indicating the texts in a pair are from the same author and the numeric author ids [1]. Out of which, 27,834 records boolean values to be true which indicates the texts for those records have been written by the same author and rest of 24,767 records have been written by different authors respectively. We studied on 20 different features which are as follows : *POS Tags count, and frequencies for Verbs, Noun, Adjective and Pronoun as well as Word 1-gram, 2-gram, 3- gram for profile length 100 and 200 and, Character 5-gram, 6-gram, 7-gram and 8-gram for profile length 100 and 200.*

Dataset	Number of Same Authors	Number of Different Authors
Ground Truth Training	24990	22350
Ground Truth Test	2844	2417
Combined Dataset	27834	24767

Table 1: Dataset statistics

5 METHODOLOGY

In the task of Authorship Verification of PAN 2020 challenge, two documents are given, and the task is to determine if these two documents are written by the same author or not by analyzing the writing style. There are several different features that can be used to distinguish the author of texts like Total Punctuation Count, Specific Punctuation Ratio, Long-sentence/ Short-sentence Ratio, Vocabulary Strength, Word N gram, Character N Gram, POS Frequency, POS Sequence Frequency, Starting POS Frequency. In this project, we are given with 52601 pairs of texts, so we have divided the dataset into 90:10 ratio of training and test data set. Then we are using dissimilarity algorithm to measure the dissimilarity between two texts using below formula [6]:

$$D(P_1, P_2) = \sum_{x \epsilon (P_1 \cup P_2)} \left(\frac{f_{P_1}(x) - f_{P_2}(x)}{\frac{f_{P_1}(x) + f_{P_2}(x)}{2}} \right)^2 \quad (1)$$

Using the formula 1, we measure the dissimilarity of two documents for one feature for a particular profile length (here we have taken profile lengths as 100 and 200). In the equation 1, we are counting the dissimilarity between two profiles P1 and P2 of the given text pairs, Where $f_{P_1}(x)$ and $f_{P_2}(x)$ are the frequencies of feature x respectively for document 1 and document 2.

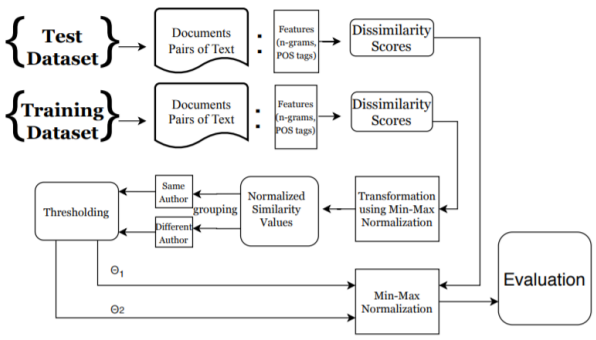

Figure 1: Proposed Methodology for Authorship verification.

5.1 Dissimilarity model and program flow

In our project, we have received only training data set from PAN 2020 challenge as a zip file. After extracting it, we have training file and truth file.

The program flow for our project is shown in the figure 1 and the code of our project is available in FimGit [1].

First we have randomly shuffled the given data pairs and divided them into 90:10 ratio into training and test dataset which we are using for our method. In our method, we are evaluating different features separately and then using evaluator provided by the PAN 2020 to calculate the performance of our method for each feature. As different profile lengths give different results, we have decided to experiment on two profile lengths 100 and 200. For each feature and profile lengths, we have calculated the dissimilarity value using the dissimilarity algorithm equation 1. After that, we have transposed the values between 0 and 1 using min-max normalization [11]. As the values are between 0 and 1, we can get the similarity values just by subtracting the dissimilarity values from 1. From the data given by the PAN 2020 challenge team, we also have the truth value of each pair of training data which tells us if the text pair is written by same author or a different one. Using this information, we have grouped the similarity values into 'same' and 'different' author category. Then by averaging these values for each group, we have threshold values for 'same' and 'different' author pair which we will use to determine the test data pair. For test data pairs, we have calculated the dissimilarity values with the same method but for min-max normalization of these values, we are using min and max values of training data set. Let us call the threshold values for different author θ_1 and for same author θ_2. As we are working on similarity, θ_2 is always greater than θ_1. Now we can conclude that the similarity values which are less than θ_1 in test data set are the different author and the values which are more than θ_2 are same author.

[1] https://git.fim.uni-passau.de/padas/20ss-tmp/team11

5.2 PAN 2020 Evaluator

In this task, we are given the evaluator by PAN 2020 team to calculate the efficiency of our model. The evaluator will compare the output file and the truth file to count the efficiency of our model based on four parameters: AUC-ROC, c@1, $F_{0.5u}$ and F1-score. The evaluator is programmed in such a way that it will count the values less than 0.5 as different author and greater than 0.5 as same author. As the evaluator is also considering the parameter c@1, which is based on the principle that non-responding response has more weightage than incorrect response, the value 0.5 will be considered as a non-response.

Now to use the given evaluator, we have transformed the similarity values of test data. We have considered the values between thresholds θ_1 and θ_2 as 0.5. As per our method, the values which are greater than θ_2 should be same author, however θ_2 in our experiment for all features lies between 0 and 0.5. So there are some values between θ_2 and 0.5 which should be considered as same author but as per the evaluation method provided by PAN2020 the values less than 0.5 will be considered as different author. So, we have mapped these values between θ_2 and 0.5 to 0.5 and 1 range to consider them as same author.

5.3 Types Of features

For our experiment, we have considered the following features.

- **Word n-gram:** In this feature, we compare the frequencies of word n-grams like unigram, bigram, trigram, etc. to measure the similarity between the two given documents.
- **Character n-gram:** In this feature, we compare the frequencies of character n-grams like character 4-gram, character 5-gram, etc. to measure the similarity between the two given documents.
- **PosTag:** In this feature, we consider different tags of the word like noun, adjective, pronoun, verb, etc. and compare their frequencies between two texts. Eg. 100 nouns with most frequency or 200 verbs with most frequency.
- **PosTagCount:** In this feature, we just compare the count of different Pos Tags between the two given documents.
- **Total Punctuation Count:** In this feature, we compare the frequencies of different punctuation symbols like comma (,), semicolon (;), question-mark (?), exclamation-mark (!), stop (.), slash (/), dash (-), colon (:) etc. between the given two documents.
- **Specific Punctuation Ratio:** In this feature, first we calculate the frequencies of different punctuation symbols and then normalize it by dividing all the frequencies by total number of punctuation symbols and use the normalized frequencies to compare the given two documents.
- **Long sentence – short sentence ratio:** In this feature, first we decide the length of long and short sentence Eg. If length of the sentence is greater than 15 words, then its long and less than 15 is short sentence. Then we compare the frequencies of both the sentences in the given two documents.
- **Vocabulary strength:** In this feature, we compare the vocabulary strength of the given texts, meaning ratio of unique words and total words.

6 EVALUATION MEASURES

In our experimental setup, we are employing PAN-20 evaluation measures that are AUC-ROC, c@1, $F_{0.5u}$ and F1-score [1]. For an evaluation, any non answers would be set to a threshold of 0.5 indicating a non decision [2]. And later on overall performance by taking the mean of all above scores.

- **AUC-ROC:** It is used for measuring performance of the classification problem at various threshold settings [10]. This score is calculated using scikit-learn [11]. Higher AUC-ROC score signifies a better performance [10] [3]. In AUC-ROC score, the non-decision is also taken into account [2].
- **c@1 :** It is an extension of accuracy measure which was introduced by Peñas and Rodrigo (2011) which is based on the principle that a non-responding response has more weightage than incorrect response [12]. It is calculated as shown below:

$$c@1 = \frac{n_{ac}}{n} + \frac{n_{ac}}{n}\frac{n_u}{n} = \frac{1}{n}(n_{ac} + \frac{n_{ac}}{n}n_u) \tag{2}$$

  where
  n_{ac} : Number of questions for which the answers is correct,
  n_{aw} : Number of questions for which the answers is incorrect,
  n_u : Number of questions not answered, and
  n : Total number of questions where n= n_{ac} + n_{aw} + n_u
  It has a good discrimination power, stability and sensitivity properties [12]. This score calculation is included in PAN 2020 [2].
- **F1-score :** It is the most popular performance measure used for classification which is calculated as **"a weighted average of the precision and recall"** [1][11]. The calculation is shown below.

$$F1 = 2 \times \frac{(precision * recall)}{(precision + recall)} \tag{3}$$

  Similarly to AUC-ROC, this score is also calculated using scikit-learn. In F1-score, the non-decision is not taken into account [2]. .
- $\mathbf{F_{0.5u}}$ **:** It is a newly proposed method introduced by the authors in [4] which is a combined form of Fβ score and c@1 score. This method overcomes the problem of c@1 where the reliable decision is not required for checking whether two texts are written by the same author [2]. For this purpose, $F_{0.5u}$ score was introduced where non answers were treated as false negatives. As $\beta = 0.5$, precision is higher than recall. The score calculation is shown below and the required calculation is provided by PAN2020 organisers [2].

$$F_{0.5u} = \frac{(1 + 0.5^2) \cdot n_{tp}}{(1 + 0.5^2) \cdot n_{tp} + 0.5^2 \cdot (n_{fn} + n_u) + n_{fp}} \tag{4}$$

  where
  n_{tp} : Number of true positives,
  n_{fn} : Number of false negatives,
  n_{fp} : Number of false positives, and
  n_u : Number of unanswered problems

7 RESULTS

We implemented our algorithm on the Postag count feature and PosTag features for profile length of 100 for verb, noun, pronoun and adjective. For profile length 100 and 200, we used word unigram, word bigram, word trigram as well as character 4-Gram, character 5-Gram, character 6-Gram, character 7-Gram, and character 8-Gram. After the successful running of evaluation on our algorithm, the results are presented in tables 2, 3, 4.

8 FUTURE WORK

In our research, we tried to implement individual features and find the optimal ones that provided us with best results. Improvement can be done on this by finding a way to combine different features effectively and observe the results by comparing them with the results of individual features. Since we chose not to manually assign the threshold value of the normalised similarity scores but instead do it automatically, some more optimistic work can be carried out on the improvement of the thresholding technique for betterment of the results.

9 CONCLUSION

In this project, we have experimented different features with different profile lengths using dissimilarity algorithm and automatic thresholding technique for authorship verification of the given two texts. From the results, we can conclude that features like character n-gram and word n-gram are quite useful to decide if two documents are written by the same author or not because each author has a unique style of writing. Also the overall result for profile length 200 is better than the result for profile length 100. Out of all the PosTag features, the result for pronoun is comparatively better than others.

ACKNOWLEDGMENTS

To PAN 2020 for providing the useful information about the authorship verification challenge, data set. To University Of Passau, for giving opportunity to work on this project and for needed guidance.

REFERENCES

[1] 2020. *Authorship Verification 2020.* Retrieved 2020-09-20 from https://pan.webis.de/clef20/pan20-web/author-identification.html

[2] 2020. *Authorship Verification 2020.* Retrieved 2020-09-20 from https://github.com/pan-webis-de/pan-code/blob/master/clef20/authorship-verification/pan20_verif_evaluator.py

[3] Taya Bazhenova. [n.d.]. *ROC and AUC, Clearly Explained!* StatQuest with Josh Starmer. https://www.youtube.com/watch?v=4jRBRDbJemM&ab_channel=StatQuestwithJoshStarmer

[4] Janek Bevendorff, Benno Stein, Matthias Hagen, and Martin Potthast. 2019. 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, 654–659. https://doi.org/10.18653/v1/N19-1068

[5] Sara Elmanarelbouanani and Ismail Kassou. 2013. Authorship Analysis Studies: A Survey. *International Journal of Computer Applications* 86 (12 2013). https://doi.org/10.5120/15038-3384

[6] Magdalena Jankowska, Evangelos Milios, and Vlado Keselj. 2014. Author Verification Using Common N-Gram Profiles of Text Documents. *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers* (Aug. 2014), 387–397. https://www.aclweb.org/anthology/C14-1038.pdf

[7] Patrick Juola and Efstathios Stamatatos. 2013. Overview of the author identification task at PAN 2013. *CEUR Workshop Proceedings* 1179 (01 2013).

[8] Moshe Koppel and Jonathan Schler. 2004. Authorship verification as a one-class classification problem. *Proceedings of the twenty-first international conference on Machine learning 2004; Banff, Alberta, Canada.* https://doi.org/10.1145/1015330.1015448

[9] Moshe Koppel and Yaron Winter. 2014. Determining If Two Documents Are Written by the Same Author. *Journal of the Association for Information Science and Technology* 65 (01 2014). https://doi.org/10.1002/asi.22954

[10] Sarang Narkhede. 2018. *Understanding AUC - ROC Curve.* Retrieved 2020-09-20 from https://towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5

[11] 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, and E. Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* 12 (2011), 2825–2830.

[12] Anselmo Peñas and Álvaro Rodrigo. 2011. A Simple Measure to Assess Non-response., Vol. 1. 1415–1424.

feature	AUC-ROC	c@1	$F_{0.5u}$	F1-score	Overall
PosTagVerb	0.571	0.633	0.655	0.697	0.639
PosTagNoun	0.572	0.641	0.662	0.694	0.642
PosTagPronoun	0.678	0.558	0.596	0.775	0.652
PosTagAdjective	0.565	0.613	0.641	0.66	0.62
PosTagCount	0.653	0.558	0.595	0.758	0.641

Table 2: Results for PosTag

feature	AUC-ROC	c@1	$F_{0.5u}$	F1-score	Overall
Word1	0.773	0.61	0.603	0.777	0.691
Word2	0.554	0.701	0.703	0.764	0.68
Word3	0.554	0.701	0.703	0.764	0.68
Character4	0.75	0.671	0.682	0.736	0.71
Character5	0.59	0.671	0.682	0.734	0.669
Character6	0.58	0.67	0.683	0.73	0.666
Character7	0.577	0.664	0.679	0.721	0.66
Character8	0.579	0.667	0.682	0.722	0.662

Table 3: Results for Word and Character N-gram for profile length 100

feature	AUC-ROC	c@1	$F_{0.5u}$	F1-score	Overall
Word1	0.783	0.664	0.674	0.74	0.715
Word2	0.564	0.714	0.712	0.774	0.691
Word3	0.586	0.689	0.699	0.742	0.679
Character4	0.774	0.677	0.685	0.745	0.72
Character5	0.592	0.68	0.688	0.744	0.676
Character6	0.562	0.677	0.687	0.74	0.667
Character7	0.569	0.686	0.694	0.746	0.674
Character8	0.591	0.693	0.701	0.747	0.683

Table 4: Results for Word and Character N-gram for profile length 200