Author Clustering with the Aid of a Simple Distance Measure

Notebook for PAN at CLEF 2017

Houda Alberts*

University of Amsterdam, the Netherlands
houda1996@hotmail.com

Abstract A simple distance measure has been applied to the author clustering problem to determine which documents are written by the same author. This simple distance measure works with the probability distribution of character sequences of a document, making it insensitive to language differences. The top most frequent features k, where k is chosen to be 300, determine the distribution where punctuation is present. Also, the uppercase letters are transformed to lowercase symbols, while a threshold of 3.0 remains for the symmetric distance score. In addition, character 2-grams are chosen due to their best outcomes. Using the BCubed F-score provided, it achieves a score of 0.54 on the training set and a score of 0.53 on the test set with a relative low MAP score. Obtaining clusters from links still shows problems.

1 Introduction

Authorship identification is an important aspect within stylometry that can be applied to many scenarios. For example, determining the author of a ransom note can save someone's life, discovering whether all the uploaded assignments of a student are classified as their own work can reduce the amount of plagiarism, but it can be also applied in arts to identify an author of an old text [5]. While some of these illustrations are based on authorship verification, finding out whether someone is the author belongs to the author clustering problem as well.

This author clustering problem [10] is the task of partitioning a given set of documents in such a way that all documents in each partition are written by the same author, and clusters are maximal with respect to this property. These sets of documents vary in genre, language, and the number of authors. Two parts are present during this task: the establishment of links between documents (denoting that two documents are written by same author) and the clustering task.

Based on the two best performing systems from [3] and [6], it can be concluded that a recurrent neural network and a simple clustering algorithm can obtain good results. Since the focal point of this paper will be language processing and feature extraction, and not machine learning, the simple distance measure from [6] will be adapted and explored further instead of the recurrent neural network from [3].

* This work was done as part of my bachelor thesis Artificial Intelligence [1] under the supervision of Maarten Marx

This paper will start with a short summary about the training data and the evaluation measures. Afterwards, the applied distance measure Spatium-L1 [6] will be discussed with the needed text processing in depth. Lastly, the results will be discussed while also concluding with alternative options for future research to obtain better performance.

2 Author Clustering Task at PAN-2017

2.1 Training data

First, the training data is explored. This data is provided by PAN [9] to apply text forensics. The data consists of problem sets in three languages: *Dutch*, *English* and *Greek* where each language is represented in two genres, reviews and articles. For these six possibilities, 10 problem sets are involved with different sizes. The size of each text snippet is also variable, meaning that there are different amounts of clusters, various authors who made a single document in a set and other quantities of (unique) words. The Greek review documents are the smallest in mean ($8021/2000 \approx 40.1$ words), while the documents for Dutch reviews are in comparison much larger ($25583/182 \approx 140.5$ words), meaning that measures focusing on only small texts for example would not be sufficient for each part of the data. All these facts can additionally be found in Table 1. The clusters size 1 column represents how many authors only contributed to one document within a problem, e.g clusters of size 1.

			Total			
Corpus	Problem Sets	Texts	Clusters size 1	Clusters	Words	Unique Words
Dutch Reviews	10	182	3	65	25583	4607
Dutch Articles	10	200	20	53	10543	3305
English Reviews	10	194	19	61	12073	3547
English Articles	10	200	18	56	10529	3425
Greek Reviews	10	200	21	61	8021	3187
Greek Articles	10	200	15	60	9824	2840
Total	60	1176	96	356	76573	20911

Table 1. Statistics training data 2017 in absolute amounts

When a program is built, it can be evaluated with the TIRA evaluation software [8] that runs the code from a virtual machine. To have the program working correctly for it to work on TIRA, it is necessary to output the results in a specific manner. The found clusters should be constructed as a nested list of documents belonging to one cluster and stored in a *clustering.json* file in the folder belonging to that problem. The links should be written to a file, in decreasing order (e.g highest first), with their corresponding scores in the *ranking.json* file.

2.2 Evaluation

For this year's evaluation, two measures are introduced. The first one is the BCubed F-score from [2] based on the regular F-score in information retrieval. What this BCubed

F-score does, is taking the average F-scores of each class to obtain an evaluation on both the complete outcome and the separate clusters. The F-score is calculated as the harmonic mean of both the precision and recall, where these are BCubed as well. This evaluation measure however, only judges the clusters that are acquired and not the links made.

To assess the performance of the created links, the mean average precision (MAP) score from [7] is determined. This technique takes the precision value for each query, a problem set in this case, and converts this to the average precision of each query. The importance of this evaluation method lies within an ordered outcome. The links that the program returns are ordered by score to have the most important scores at the top and the lower ones at the bottom. The MAP handles this valuable information by calculating the precision based on whether it appears at the right place as well.

3 Our Method

3.1 Text Processing

The first step to obtain the character N-gram features is based on text processing. The text is lowercased to reduce the amount of possible features that can be extracted. All other symbols, including punctuation, are left within the text, just as the more frequent terms. Lemmatization, stemming or any other method to smooth the amount of words into more general version of the word within a text are not applied.

Mainly the more frequent terms within a text, the function words, say more about the writing style of a person [5]. Keeping this information in mind, more frequent terms should not be removed or under-weighted. Features are then extracted by using character N-grams. These character N-grams do include punctuation and spaces and have length N. For example, the sentence: *Hello, have you seen Alice?* would also yield *ello,*, *o, ha* and *Alice* as character 5-grams. The submitted software uses character 2-grams since those features yielded the best result out of all character N-grams where N ranges from 1 to 6. Unique character N-grams can remain within these features and could possibly be disregarded by choosing only the top k most frequent character N-grams in the text. [6] proposes, with evidence from Burrows and Savoy, that values between 200 and 300 for k yield the best outcomes, so these will be included as probable values. This results in frequencies that are then converted into probabilities by the relative frequency, as mentioned in (1) where $P(t)$ is the chance of feature t and T is the set of all features. Due to the normalization over all features, the sum of all elements within the probability distribution adds up to 1.

$$P(t) = \frac{\text{frequency}_t}{\sum_{t' \in T} \text{frequency}_{t'}} \tag{1}$$

3.2 Simple Distance Measure

When the documents are converted to probability distributions as discussed, the simple distance measure from [6], Spatium-L1, can be applied. It takes the absolute differences

of the two vectors element-wise and sums it up, as shown in (2), where the range k is defined by the features from the first probability distribution.

$$\Delta(P_{\text{doc}_1}, P_{\text{doc}_2}) = \Delta_{12} = \sum_{i=1}^{n} |P_{\text{doc}_1}(i) - P_{\text{doc}_2}(i)| \tag{2}$$

After obtaining these summations, these scores are transformed to standard deviations, where a high standard deviation score yields more evidence that the two documents are written by the same author meaning that it becomes a similarity measure instead of dissimilarity measure. This is done by first calculating the average of each document, which is the average distance from a document to all other documents within the problem set. The standard deviation of this document is then also determined by comparing all the distances from a single document to all other documents with the average distance for that single document. For example, if $\Delta_{12} = 40$, the average from document 1 to all other documents is 50 and the standard deviation for this document has the value 5, the score can be calculated as follows: $\frac{40-50}{5} = 2$, making this the new updated standard deviation score for document 1 and document 2.

Since these standard deviations are based on the features from the probability distribution of the first document (e.g the most frequent top k features), the values for $\Delta(\text{doc}_1, \text{doc}_2)$ are not the same as for $\Delta(\text{doc}_2, \text{doc}_1)$ (i.e., $\Delta_{12} \neq \Delta_{21}$) making these scores and the scores converted to standard deviations non-symmetric. This shows that using only one score, e.g one direction, would not be sufficient enough to say something about the link. Therefore to keep the simple characteristic of this technique, the symmetric standard deviation score is computed by the addition of both the non-symmetric scores of the two documents. For instance, if standard deviation $\text{score}_{12} = 3$ and standard deviation $\text{score}_{21} = 2$, the symmetric score will be $3 + 2 = 5$. Lastly, this symmetric standard deviation score will be compared against an arbitrary threshold that will indicate whether the two documents are written by the same author if the score is higher than the threshold, otherwise they are not written by the same author. Afterwards these symmetric standard deviation scores are scaled down to a value between 0 and 1 with the aid of the maximum and minimum obtained scores as shown in 3.

$$score_i = \frac{\text{distance}_i - min(\text{distance})}{max(\text{distance}) - min(\text{distance})} \tag{3}$$

3.3 Clustering

When Spatium-L1 has been applied on the data, several links or none at all are found. These are then used to cluster the documents to find which sets of documents are written by the same author. This done with the use of connected components. All the possible documents are added as nodes within a graph and then all the links found by the measure will be given to the graph as well. The connected components option then returns clusters of documents where a cluster contains documents that are all linked by each other with at least one link. For instance, if document 1 and 2, document 2 and 3, and document 3 and 4 are all linked, they would form the cluster containing document 1, 2,

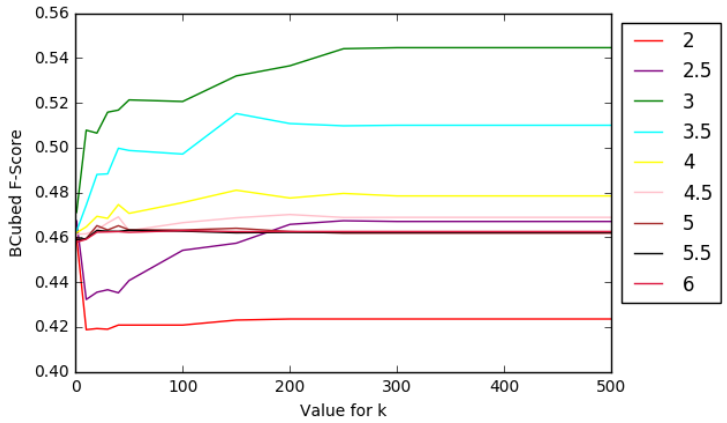
3 and 4.

Another example can be found in Figure 1, where it can be seen that the upper-right document is connected to the upper-left document by documents in between, which still indicates a connection making the purple cluster. This shows that every link will considered equally strong, while it may be that the connection between the upper-right document and the middle document is not as strong as it should be to be considered a part of the cluster which is its downside.

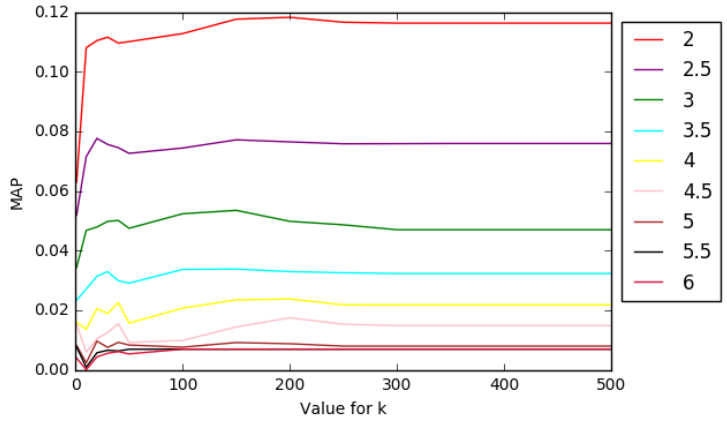
Figure 1. The connected components clustering option; all connected documents make a cluster indicated with a single color

4 Results

When running the simple distance measure on the training data, the highest possible BCubed F-score is achieved by setting k equal to 300 and the threshold to 3.0 with a score of 0.54, while using character 2-grams. Different values for N (1 till 6) were explored for the character N-grams, but for $N = 2$, the best results were obtained. The results for the other character N-grams are therefore disregarded. The results, the BCubed F-Score and MAP value, can be found in Figures 2a and 2b respectively for different values of k and for different threshold between 2.0 and 6.0. It can be concluded that the best outcomes for the BCubed F-Score are found when $k = 300$ and the threshold is 3.0 as mentioned before, not too high but not too low either. However, the MAP score is maximized for a different threshold, which is 2.0. This can be explained due to a lower threshold being more lenient with assessing links and returning more links as a result yielding a better performance on this part. Although, this is more beneficial for the MAP score, it is not for the F-Score, which should have a higher threshold to sustain sufficient outcomes which is why the software has been submitted with with the parameters achieving the best BCubed F-score.



(a) BCubed F-Scores



(b) MAP values

Figure 2. BCubed F-Scores and MAP values for different k values per threshold for character 2-grams on the training set

For the outcomes on the test data, the performance does differ greatly between all the participants as shown in Table 2. Whereas the BCubed F-score of my software shows a good result marked in pink, the MAP score is quite low showing there can be more improvement on that part compared to the participant with the best result in yellow. It does however show BCubed F-scores that are way lower than last year, which may be due to the differences in those datasets. Last year, the dataset contained larger problem sets and the text was also much longer (three paragraphs for example) meaning that more evidence could be found about someone's writing style in the text, while this was not the case this year with smaller problem sets and less text in the documents. This

shows there should be a robust measure or algorithm to solve this problem to work with both kind of datasets.

Participant	BCubed F-score	MAP	Runtime
Alberts	0.53	0.04	00:01:45
Gómez-Adorno et al.	0.57	0.46	00:02:05
García et al.	0.56	0.38	00:15:49
Halvani & Graner	0.55	0.14	00:12:25
Karaś	0.47	0.13	00:00:26
Kocher & Savoy	0.55	0.40	00:00:41

Table 2. Outcomes PAN competition 2017 on test set

5 Conclusion & Discussion

To conclude, using the top 300 most frequent terms of each document and character 2-grams, a BCubed F-score of 0.54 is achieved on the, provided, training data with a threshold of 3.0 whereas the same parameters obtain a BCubed F-score of 0.53 on the test set. While this method does show some performance, it is still an inefficient result. Based on the current technique, there are still opportunities for further enhancements. For example, only character N-grams, where N lies between 1 and 6, are tested on performance. Higher values for N or word grams may improve the outcome. Another improvement lies within the text processing, that currently only consists of converting all the text to lowercase. Using uppercase characters, replacing unique words with a specific symbol or removing punctuation may all be improvements on the data. Next, the parameters could be separately set for each language and genre. Linguistic features of a language can influence the recognition of writing styles, meaning that these separate parameters could improve the outcome. However, this could not be tested for this submission due to time constraints.

Lastly, the clustering aspect, from the found links, is not the most optimal solution. By applying the connected components method blindly, weak links may bind two separate clusters wrongly. An alternative lies within the possibility to using weighted links when extracting the clusters such as the Louvain modularity from [4]. By doing so, weak links between strong clusters can be discarded to obtain better performance.

Since this task is also the topic of my bachelor thesis, I continued with the research after the submission trying some of the previous mentioned alternatives to improve the results. Using an other more sophisticated distance measure, the Jensen-Shannon Divergence, transpired to work much better obtaining BCubed F-scores around 0.53 and MAP scores exceeding 0.3. Word N-grams are not an improvement, while the Louvain modularity does do more sophisticated clustering, but the found links must be more precise. The text processing alternatives also do improve the quality of the links, meaning a higher MAP score is achieved. With these results, it can also be concluded that more sophisticated distance measures are worth looking into.

References

1. Alberts, H.: Authorship Identification by Author Clustering (Unpublished bachelor thesis) (2017)
2. Amigó, E., Gonzalo, J., Artiles, J., Verdejo, F.: A comparison of extrinsic clustering evaluation metrics based on formal constraints. Information Retrieval 12(4), 461–486 (Aug 2009)
3. Bagnall, D.: Authorship clustering using multi-headed recurrent neural networks. In: Working Notes of CLEF 2016 - Conference and Labs of the Evaluation forum. pp. 791–804. Citeseer (2016)
4. Blondel, V.D., Guillaume, J.L., Lambiotte, R., Lefebvre, E.: Fast unfolding of communities in large networks. Journal of Statistical Mechanics: Theory and Experiment 10, 10008 (Oct 2008)
5. Kestemont, M., Stronks, E., de Bruin, M., de Winkel, T.: Van wie is het wilhelmus? Amsterdam University Press (2016)
6. Kocher, M.: Unine at clef 2016: Author clustering. In: Working Notes of CLEF 2016 - Conference and Labs of the Evaluation forum. pp. 895–902. Citeseer (2016)
7. Manning, C.D., Raghavan, P., Schütze, H., et al.: Introduction to information retrieval, vol. 1. Cambridge university press Cambridge (2008)
8. Potthast, M., Gollub, T., Rangel, F., Rosso, P., Stamatatos, E., Stein, B.: Improving the Reproducibility of PAN's Shared Tasks: Plagiarism Detection, Author Identification, and Author Profiling. In: Kanoulas, E., Lupu, M., Clough, P., Sanderson, M., Hall, M., Hanbury, A., Toms, E. (eds.) Information Access Evaluation meets Multilinguality, Multimodality, and Visualization. 5th International Conference of the CLEF Initiative (CLEF 14). pp. 268–299. Springer, Berlin Heidelberg New York (Sep 2014)
9. Potthast, M., Rangel, F., Tschuggnall, M., Stamatatos, E., Rosso, P., Stein, B.: Overview of PAN'17: Author Identification, Author Profiling, and Author Obfuscation. In: Jones, G., Lawless, S., Gonzalo, J., Kelly, L., Goeuriot, L., Mandl, T., Cappellato, L., Ferro, N. (eds.) Experimental IR Meets Multilinguality, Multimodality, and Interaction. 8th International Conference of the CLEF Initiative (CLEF 17). Springer, Berlin Heidelberg New York (Sep 2017)
10. Tschuggnall, M., Stamatatos, E., Verhoeven, B., Daelemans, W., Specht, G., Stein, B., Potthast, M.: Overview of the Author Identification Task at PAN 2017: Style Breach Detection and Author Clustering. In: Cappellato, L., Ferro, N., Goeuriot, L., Mandl, T. (eds.) Working Notes Papers of the CLEF 2017 Evaluation Labs. CEUR Workshop Proceedings, CLEF and CEUR-WS.org (Sep 2017)