Local n-grams for Author Identification
Notebook for PAN at CLEF 2013

Robert Layton, Paul Watters, and Richard Dazeley

Internet Commerce Security Laboratory
University of Ballarat
r.layton@icsl.com.au, p.watters@ballarat.edu.au, r.dazeley@ballarat.edu.au

Abstract Our approach to the author identification task uses existing authorship attribution methods using local n-grams (LNG) and performs a weighted ensemble. This approach came in third for this year's competition, using a relatively simple scheme of weights by training set accuracy. LNG models create profiles, consisting of a list of character n-grams that best represent a particular author's writing. The use of a weighted ensemble improved upon the accuracy of the method without reducing the speed of the algorithm; the submitted solution was not only near the top of the leaderboard in terms of accuracy, but it was also one of the faster algorithms submitted.

The authorship identification task at PAN 2013 was a variation on a standard authorship analysis task of authorship attribution. In authorship identification, we have a training set of documents from the same author and a test document of unknown authorship. The task is to determine whether the author of the training documents was the one that wrote the test document. This task is different from authorship attribution in a few ways. First, we cannot simply take a 'best guess' whereby we find the best matching author. A decision on match or no match must be made, similar to the open set problem of authorship attribution, whereby the actual author may not be in the candidate set. Second, we have no point of reference to compare the similarity of author to document. In other words, we cannot know relatively if two profiles are similar and must therefore find algorithms that are able to know *absolutely* if two profiles match. Third, specifically for this task, the number of documents was small. Most problems in this task had just three documents from the same author, reducing the ability to determine variance.

1 Local n-grams

Local n-gram models were first proposed by [9] in 2003, who introduced the Common n-grams algorithm (CNG). CNG was developed for authorship attribution, working by profiling authors and documents based on the list of the top L character n-grams. Character n-grams had been used in authorship studies prior, but previous methods used a 'global' set of n-grams. CNG instead used 'local' profiles, whereby an author's (or document's) profile consists only of the most frequent character n-grams *for that author (or document).*

Authorship attribution for a test document is performed by comparing the profile for the document with the profiles of each of the candidate authors using equation 1 (where $P_i(x)$ is the frequency of n-gram x for a given profile). The most similar profile is considered the author of the test document.

$$d_K(P_1, P_2) = \sum_{x \in X_{P_1} \cup X_{P_2}} \left(\frac{2 \cdot (P_1(x) - P_2(x))}{P_1(x) + P_2(x)} \right)^2 \tag{1}$$

[5] identified that a much simpler algorithm was capable of comparable results, called Source Code Author Profiling (SCAP). Rather than use the frequency of the most frequent n-grams, two profiles are compared simply by the size of the intersection of the n-grams in the profiles. While this does reduce the accuracy, the net result is often quite small and the algorithm is significantly faster. As an empirical result, our code for CNG performs a profile comparison in about three milliseconds, while for SCAP it is less than one millisecond (our testing framework does not give better precision).

Further variations were developed by [14], who focused on variations of CNG for unbalanced datasets. The first variation, $d1$ has a single subtle change, using only the n-grams in the test document for comparison. The second variation, $d2$ proposed in [14] is to use a training corpus profile, called P_C, which is composed of all the documents in the training set. The distance is given as the dot product of the $d1$ distance from the document to author and the $d1$ distance of the document to the language profile. These alterations to CNG proved to be more accurate than ‘standard’ CNG when there was a limited number of documents for an author. However when an adequate number of documents per author is reached, the results reverse and standard CNG, using equation 1, proves to be more accurate.

Another variation was developed by [13] named Recentred Local Profiles (RLP). While the distance metric was different from CNG and SCAP, using a variation of the cosine distance metric, the main difference was the use of a language default value, similar to [14]. Rather than profile using the most frequent n-grams for an author or document, the *most distinctive* n-grams are chosen. An n-gram is more distinctive than another if the absolute distance to the expected value for the language is higher. The expected value for an n-gram can be calculated from a corpus of text in the language, and has the benefit of being able to use unlabelled data. RLP was shown to have a high quality on many datasets, including the AAAC corpus [7].

LNG methods have been used for a significant number of tasks. Some successful applications include authorship of twitter posts [11], student coding essays [2,3], software forensics [6], internet relay chat logs [10], malware profiling [1] and has been extended into automated unsupervised applications [12]. LNG based approaches are highly applicable to any language, as the notion of a character is quite prolific. In cases where a character can be hard to define, often examining the bytes of a representation (such as unicode) can provide an adequate substitution [4].

2 Methodology

The LNG methods described in the literature are mostly developed for authorship attribution and not authorship identification. Intuitively, we created an algorithm that identifies if the test document compares to the training set in the same way as documents in the training set compare to each other - i.e. if it looks like a training document, it was probably written by the same person. To convert the methods to an authorship identification task, we applied the following procedure.

1. Calculate the pairwise distance between all documents in the training set and call this distribution du.
2. Calculate the distance between the test document and each of the training documents, and call this distribution dt.
3. Compute the number of elements in dt that are less than the mean of du and call this p.
4. If $p < t$ for some t (set to 0.5 by default), then the test document is considered a match. Otherwise, it is not a match.

Formally, a 'problem' is defined as a set of documents from the same author (the training set) and a document of unknown authorship (the test document). The aim is to determine if the author of the training set wrote the test document.

The ensemble method was inspired by [8], which used a weighted voting model with local n-grams, including a 'reject' option, indicating if no match was found. In that work, a large number of CNG models were used with different parameters, and the weights were assigned based on the ratio $r = 1 - \frac{a}{b}$, where a is the distance between the test profile and the best matching author profile and b is the distance to the second best. Note that this application is valid for authorship attribution tasks, and not for authorship identification, as we only have one candidate author.

Instead, the method we used was to assign each model a weight according to its accuracy in the training set. Any model with a weight of less than 0.4 is reduced to 0. The reason for this is that in the author identification task, there are only two possible answers - match or no match. This gives a baseline accuracy of 0.5, assuming the test dataset contains equal numbers of problems of each type. We decided that models scoring under 0.4 were a hindrance rather than a help.

To summarise, out approach was to use a set of LNG models and ask each whether they thought the document was from the same author or not. This used the approach given in section 2. Each model's prediction was weighted according to their accuracy in the training set, with models scoring less than 0.4 given a weight of 0. The weights for both options (match or no match) were summed and the option with the highest weight was chosen as the final answer. The models included in the ensemble were the CNG, SCAP, RLP, d1 and d2 models. The parameters were $n \in \{3, 4, 5\}$ and $L = 1000$ for each of the five models. We hypothesise that a larger number of models would have improved accuracy, but was not able to test this hypothesis for this competition.

3 Results

The competition organisers released a set of problems for training, which were used in a cross-fold methodology to evaluate the algorithms. The results for each fold are given in table 3 and the overall accuracy was 71.25%. For the competition, the algorithm achieved and f_1-score of 0.671, which placed it in third. In addition, the algorithm was the forth fastest in the competition.

Fold	Mean	1	2	3	4	5	6	7	8	9	10
Accuracies	0.71250	1.000	1.000	0.333	1.000	0.500	0.333	1.000	0.333	1.000	0.625

Table 1. Results from each fold using weighted ensemble.

References

1. Alazab, M., Layton, R., Venkataraman, S., Watters, P.: Malware detection based on structural and behavioural features of api calls (2010)
2. Burrows, S.: Source Code Authorship Attribution. Ph.D. thesis, RMIT University (2010)
3. Burrows, S., Uitdenbogerd, A.L., Turpin, A.: Application of information retrieval techniques for source code authorship attribution. In: Database Systems for Advanced Applications. pp. 699–713. Springer (2009)
4. Frantzeskou, G., Gritzalis, S., Macdonell, S.G.: Source code authorship analysis for supporting the cybercrime investigation process. In: Proceedings of the first International Conference on e-business and Telecommunications Networks (ICETE04), Vol. pp. 85–92 (2004)
5. Frantzeskou, G., Stamatatos, E., Gritzalis, S., Chaski, C.E.: Identifying authorship by byte-level n-grams: The source code author profile (SCAP) method. Int. Journal of Digital Evidence 6 (2007)
6. Frantzeskou, G., Stamatatos, E., Gritzalis, S.: Supporting the cybercrime investigation process: effective discrimination of source code authors based on byte-level information. In: E-business and Telecommunication Networks, pp. 163–173. Springer (2007)
7. Juola, P.: Ad-hoc Authorship Attribution Competition. In: Proceedings of 2004 Joint International Conference of the Association for Literary and Linguistic Computing and the Association for Computers and the Humanities (ALLC/ACH 2004) (2004)
8. Kešelj, V., Cercone, N.: Cng method with weighted voting. In: P. Joula, Ad-hoc Authorship Attribution Competition (ALLC/ACH 2004), Göteborg, Sweden (2004)
9. Kešelj, V., Peng, F., Cercone, N., Thomas, C.: N-gram-based author profiles for authorship attribution. In: Proceedings of the Pacific Association for Computational Linguistics (2003)
10. Layton, R., McCombie, S., Watters, P.: Authorship attribution of irc messages using inverse author frequency. In: Cybercrime and Trustworthy Computing Workshop (CTC), 2012 Third. pp. 7–13. IEEE (2012)
11. Layton, R., Watters, P., Dazeley, R.: Authorship attribution for twitter in 140 characters or less. In: 2010 Second Cybercrime and Trustworthy Computing Workshop. pp. 1–8. IEEE (2010)
12. Layton, R., Watters, P., Dazeley, R.: Automated unsupervised authorship analysis using evidence accumulation clustering. Natural Language Engineering, Available on CJO (2011)
13. Layton, R., Watters, P., Dazeley, R.: Recentred Local Profiles for Authorship Attribution. Journal of Natural Language Engineering (2011), available on CJO 2011
14. Stamatatos, E.: Author identification using imbalanced and limited training texts. In: Database and Expert Systems Applications, 2007. DEXA'07. 18th International Workshop on. pp. 237–241. IEEE (2007)