A Behavioral Biometrics based Authentication Method for MOOC's that is Robust against Imitation Attempts

Markus Krause
Leibniz University
Hannover, Germany
markus@hci.uni-hannover.de

ABSTRACT
Ensuring authorship in online taken exams is a major challenge for e-learning in general and MOOC's in particular. In this paper, we introduce and evaluate a method to verify student identities using stylometry. We present a carefully composed feature set and use it with a K-Nearest Neighbor algorithm. We demonstrate that our method can effectively authenticate authors and is robust against imitation attacks.

Author Keywords
Author Authentication; Human Factors; Massive Open Online Courses; e-Learning

ACM Classification Keywords
H.5.m. Information interfaces and presentation; K.3.1 Computer Uses in Education; K.3.2 Computer and Information Science Education

INTRODUCTION
Exams in MOOC's become more open even to the point of free text submissions graded by peer reviews [11]. Verifying a student's identity is a crucial aspect of such free text online exams. The behavioral biometrics of stylometry is a possible solution to this challenge. Stylometry attributes authorship using features of literary style such as sentence length, vocabulary richness, frequencies of words, word lengths, and so on. The benefit of stylometry is that the authentication information is an inherent part of the text and the method does not require any further information. With carefully chosen features, it is a complex task to imitate a writing style with a computational system. Altering features such as the grammatical structure of a sentence without changing the meaning of the text seem to be challenging.

In this paper, we propose a new stylometric method that uses a well-balanced feature set and an instance based classifier to perform author authentication. We illustrate the feasibility of this method to be suitable for student authentication in MOOC's. Instance based classifier already showed excellent results with keystroke dynamics and in attempts to scale stylometry to hundredths of thousands of authors [5]. We will demonstrate that the combination of well-designed feature sets and the K-Nearest Neighbor classifier is superior to other current approaches.

L@S 2014, March 4–5, 2014, Atlanta, Georgia, USA.
ACM 978-1-4503-2669-8/14/03.
http://dx.doi.org/10.1145/2556325.2567881

CORPUS
A particular challenge for our approach is to find a suitable corpus that allows comparing our approach and contains samples of imitation attempts. While there are many corpora with known author information, we need a corpus with authors try to imitate another. A suitable data set is the Extended Brennan-Greenstadt (EBG) Corpus [3]. The corpus was created using Amazon's Mechanical Turk (AMT) platform. The contributors, which participated in the Brennan-Greenstadt experiment, have various backgrounds but at least some college education. Each contributor submitted a sample writing of at least 6500 words. Additionally as we use the same corpus we can compare our results with those reported by Brennan et al. [3]. Each sample in the corpus is from a formal source, such as essays for school, reports for work, and other professional and academic correspondence. The samples therefore already have similarity or indeed are submissions for an exam. The corpus also contains a text from each author in which she tries to imitate another author's style. For this task, the contributors got a 2500-word sample from "The Road" by Cormac McCarthy to model their passage after. The contributor's task was to narrate their day from rise on using third-person perspective. This is also similar to the events in the sample text.

FEATURE EXTRACTION
We extracted different feature sets from the corpus. Other approaches use features a machine could imitate, for instance digits. An algorithm can easily detect fractional numbers and add additional numbers to better resemble another author e.g. altering 0.98 to 0.982. This alteration would go unnoticed, as it does not change the meaning of the text. Whitespaces such as line breaks, tabs, and space are also vulnerable to machine based imitation. The individual features are described below.

Character Frequency
The relative frequency of individual characters. This feature set contains the relative frequency of a-z and A-Z.

Word Length Frequency

The relative frequency of word length. In some rare cases the part of speech tagger was not able to filter certain artifacts e.g. long numbers, some e-mail addresses (without the @ sign). This results in particular long words. To filter such elements we only use words of up to 20 characters.

Sentence Length Frequency

The relative frequency of sentence length. Similar to the word length feature we filter out overly long sentences longer than 35 words. The feature set is for obvious reasons very sensitive to small data sets. We use this feature set as we can assume in the explained scenario to have larger data sets. Training and test sets should contain at least 80 sentences when used with the classifiers proposed in this work.

Part of Speech Tag Frequency

For this feature set we use the Penn Treebank part of speech tag set. We use the Natural Language Toolkit (NLTK [2]) python library to extract these tags from a corpus. We calculate the relative frequency of each tag.

Word Specificity Frequency

The specificity of words used by an author is a discriminating feature. To our knowledge this has not been used for stylometry yet. To estimate the specificity of a word we use *wordnet*. The algorithm calculates the distance between each word and the root node of *wordnet*. The algorithm calculates the relative frequency of each depth. The depth is limited to 20.

MODEL LEARNING

For our experiment we use the instance based machine learning algorithm *K-Nearest Neighbor (KNN)* [1]. The KNN algorithm selects the k closest samples of the training set for each given test sample. The algorithm then determines the class of the test sample by counting the found train samples of each class. The algorithm is most often used with a weighting factor for each test sample. Commonly the inverse distance $\left(\frac{1}{d}\right)$ between neighbor and test sample. We use the *WEKA* [4] implementation of the KNN algorithm for our experiment. To prepare the data from the EBG corpus we split it into a train and a test set of equal size. We extract the described features and generate a vector for each sentence. Afterwards we group all vectors by their author. Through bootstrapping we aggregate samples for each author from these groups in both sets.

ROBUSTNESS AGAINST IMITATION

We want to know how robust our method is against attempts to imitate another author. As explained above each author was asked to imitate the author Cormac McCarthy. Authors had a passage of 2500 words after which they modelled their own text. We train one model for McCarthy using this text and the training data from the 45 authors of the EBG corpus. It is very likely that the author trying to imitate another is not in the database. Therefore, we exclude this author from the train set.

To test the trained model we use another text sample of ~2500 words from "The Road" written by McCarthy and the imitation samples from the author previously excluded. The imitations had 50 sentences (663 words) on average. We repeat the process for each author. To make our experiment comparable to the experiment done by Brennan et al. [3] we repeat this experiment 1000 times with different sets of 40 authors out of the initial 45. We also did the same experiment without removing the imitating author from the train set. Figure 1. shows the success rates of the imitation attacks for both experiments compared to the success rates reported by Brennan et al. [3].

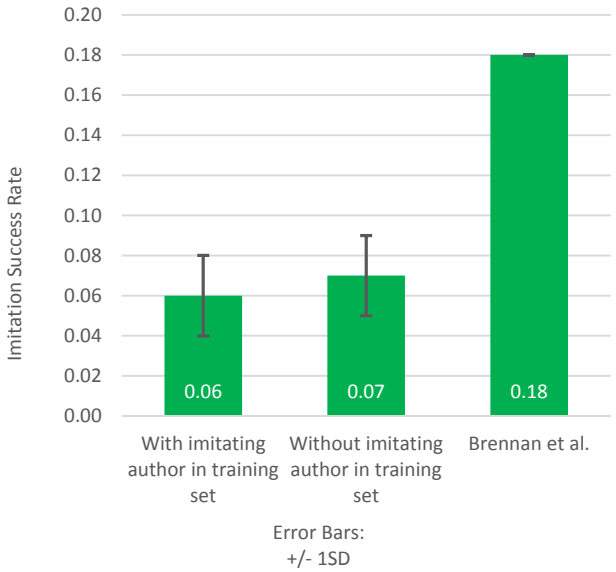

Fig 1. Success rates for the imitation experiment. Brennan et al. [3] did not report an SD for the experiment.

REFERENCES

1. Aha, D., Kibler, D., and Albert, M. Instance-based learning algorithms. *Machine learning 6*, (1991), 37–66.
2. Bird, S., Klein, E., and Loper, E. *Natural language processing with Python*. O'Reilly Media Inc., 2009.
3. Brennan, M., Afroz, S., and Greenstadt, R. Adversarial stylometry. *ACM Transactions on Information and System Security 15*, 3 (2012), 12:1–22.
4. Frank, E., Hall, M., Holmes, G., et al. Weka-A Machine Learning Workbench for Data Mining. In O. Maimon and L. Rokach, eds., *Data Mining and Knowledge Discovery Handbook*. Springer US, Boston, MA, 2010, 1269–1277.
5. Narayanan, A. and Paskov, H. On the feasibility of internet-scale author identification. *IEEE Symposium on Security and Privacy*, IEEE Press (2012).