DIGITAL FORENSIC RESEARCH CONFERENCE

ChunkedHCs Algorithm for Authorship Verification Problems: Reddit Case Study

By:
Anh Duc Le (Munster Technological University and Rigr AI),
Justin McGuinness (Munster Technological University),
and Edward Dixon (Rigr AI)

From the proceedings of

The Digital Forensic Research Conference

DFRWS USA 2021

July 12-15, 2021

Forensic Science International: Digital Investigation 37 (2021) 301185

Contents lists available at ScienceDirect

Forensic Science International: Digital Investigation

journal homepage: www.elsevier.com/locate/fsidi

DFRWS 2021 USA - Proceedings of the Twenty First Annual DFRWS USA

ChunkedHCs algorithm for authorship verification problems: Reddit case study

Anh Duc Le [a, b, *], Justin P.L. McGuinness [a], Edward Dixon [b]

[a] Department of Mathematics, Munster Technological University, Cork, Ireland
[b] Rigr AI, Cork, Ireland

ARTICLE INFO

Article history:

Keywords:
Authorship verification
Higher criticism
HC-Based similarity algorithm
ChunkedHCs
Reddit

ABSTRACT

Cybercrime can be associated with undisclosed social media accounts deliberately used to conduct unethical or illegal activities such as cyberbullying, fraudulent transactions, human trafficking, etc. The objective of this paper is to identify whether two social media accounts belong to the same person by examining the accounts' writing, i.e. comments and posts. To that end, this preliminary study introduces a new algorithm, ChunkedHCs, specifically designed for the authorship verification task to decide whether a pair of texts are written by the same person. In the domain of machine learning and deep learning, there have been previous authorship verification approaches, which often involve complex feature selections or sophisticated pre-processing steps due to the complexity of topic heterogeneity. Such limits provide motivations to seek a simpler yet more robust approach that could offer competitive verification ability. ChunkedHCs is based on the statistical testing Higher Criticism (Donoho and Jin, 2004) and the HC-based similarity algorithm (Kipnis, 2020a & 2020b) (Kestemont et al., 2020). Using Reddit users' data, ChunkedHCs offer a promising performance with an accuracy of 0.94 and an F1 of 0.9381 for texts between 29,000 and 30,000 characters. It is speculated that the algorithm could also be highly applicable to identify if two accounts are used by the same person for other social media platforms such as Facebook, Twitter and even dark web forums. Various avenues of further research on ChunkedHCs are also proposed.

1. Introduction

1.1. Context of authorship verification

According to Facebook's quarterly report in 2019, 275 million accounts were duplicated, comprising 11% of the monthly active users. Furthermore, 5% of accounts were classified as “false” or “undesirable”, totalling 137 million users (Facebook, 2019). These figures might imply potential threats, namely internet users who hide or fake their true identities to manipulate the cyberspace in order to advance their interests, to the extent that such activities could be considered illegal. They include scam, online fraud, human trafficking, drug dealing, etc. The problem accordingly requires urgent and effective measures of identifying authorship for such accounts.

Responding to the challenge, the objective herein is to identify whether two social media accounts belong to the same person by examining comments and posts from the two accounts. It is the authorship task of identifying whether two pieces of texts are written by the same person. According to Stamatatos et al. (2014), authorship verification methods are either intrinsic or extrinsic. Intrinsic methods refer to deciding whether a pair of texts are written by the same person, without having additional texts by other authors. By contrast, extrinsic methods use external documents from other authors to make decisions. Within the scope of this study, only intrinsic verification methods are examined.

The motivation of this work lies in the broad application of the authorship verification task for digital forensics and cybersecurity. In response to the abuse of the cyberspace's freedom, the study will enrich research in the area and enable academic researchers, industrial practitioners and legal enforcement officers to choose appropriate authorship verification tools.

* Corresponding author. Department of Mathematics, Munster Technological University, Cork, Ireland.
E-mail addresses: dukele35@gmail.com, duke@rigr.ai (A.D. Le).

https://doi.org/10.1016/j.fsidi.2021.301185

1.2. Previous attempts at authorship verification

Uncovering Plagiarism, Authorship, and Social Software Misuse (PAN) is a scientific community for academic researchers and industrial practitioners to study digital text forensics and stylometry. Since 2009, PAN has hosted shared tasks in plagiarism detection, Wikipedia vandalism detection, authorship analysis, among others. Particularly, there have been four specific shared tasks in authorship verification in 2013, 2014, 2015 and 2020. The best intrinsic approaches in these years are:

- K-nearest neighbour (KNN) estimation by Halvani et al. (2013),
- Classification and Regression Trees (CART) by Frery et al. (2014),
- Differential features based on random forest by Bartoli et al. (2015) and,
- Deep Bayes factor scoring by Boenninghoff et al. (2020).

Overall, the first three approaches are incorporated using machine learning algorithms; meanwhile, the last one features a statistical modelling built upon a deep learning architecture. The similarity among these approaches is the complex feature selections and multiple pre-processing steps. The first, second and third approaches have countless possibilities of feature representations, so it is challenging to attain optimal feature selections. The last approach encounters a relatively long runtime due to its complex architecture. Although those approaches yield competitive results in the PAN's shared tasks, the mentioned limits are the motivations for this study searching for a simpler technique, without compromising the authorship verification ability.

1.3. Novelty of this study

This paper introduces a new algorithm called ChunkedHCs, which is specifically designed for the authorship verification task. It is the intrinsic verification approach. ChunkedHCs' foundations are mainly inherited from the statistical testing Higher Criticism (HC) (Donoho and Jin, 2004), especially the HC-based similarity algorithm (Kipnis, 2020a & 2020b) (Kestemont et al., 2020) with some modifications. Generally, ChunkedHCs offer a more straightforward and simpler work procedure in comparison with machine learning and deep learning approaches. Firstly, there is no need to pre-process the text inputs as ChunkedHCs considers all text elements, including characters, punctuations and special characters. Secondly, ChunkedHCs are not influenced by different topics and genres, implying that the algorithm is highly versatile and applicable to various contexts. Thirdly, the output is a similarity probability, which is directly derived from a pair of texts without having additional information or datasets. Finally, the computation is much less intense than deep learning or machine learning approaches, since no training-data phase is involved.

1.4. Research questions

This paper will describe the concept of ChunkedHCs and attempt to address the following research questions:

- How are ChunkedHCs employed for the authorship verification task?
- How do ChunkedHCs perform on Reddit users' data?

1.5. Organisation of the paper

Correspondingly, this paper will be constructed in the following sections. Section 2 discusses successful attempts of performing authorship verification in previous PAN's shared tasks. Section 3 presents the theoretical foundations of ChunkedHCs. Section 4 explains ChunkedHCs step-by-step. Section 5 reports the results of applying ChunkedHCs for Reddit users' data. Section 6 evaluates ChunkedHCs' strengths and its limits. Finally, Section 7 gives the answers to the research questions together with future work.

2. Related work

This section summarises the four best intrinsic verification approaches in the PAN's shared tasks in 2013, 2014, 2015 and 2020.

2.1. KNN estimation (2013)

Given a set of training documents whose author is A, the task is to decide if the author of an unknown document is also A. Halvani et al. (2013) proposed a three-step verification process. Firstly, documents are pre-processed via normalisation and noise reduction. Secondly, pre-processed documents are converted into feature vectors, which are constructed from twelve feature categories. Then, style deviation scores between the unknown document and the training documents are derived by applying a KNN classifier for the feature vectors. Finally, the final decision is made by a majority vote from all decisions derived from style deviation scores.

Although Halvani et al. (2013) achieved the second-best result in the shared task, the approach still faces certain drawbacks. Firstly, the choice of feature categories would directly affect accuracy and runtime. Secondly, the optimal choice of feature categories was also yet to be found, due to the vast feature space. Thirdly, the approach is affected by topic heterogeneity that the verification results tend to be varied.

2.2. CART decision trees (2014)

The task definition in 2014 is the same as the PAN's shared task in 2013. Frery et al. (2014) transform the documents into eight representation spaces based on different text features, such as words, punctuation marks, vocabulary, etc. There are two attributes, i.e. count and mean, derived from each representation space. A global count attribute for all eight representation spaces is also calculated. Taken together, there are 17 attributes characterising a verification problem featuring a set of documents whose author is A and an unknown document whose author claims to be A. CART decision trees are then used to perform classification, i.e. whether the author of the unknown document is A, based on such attributes. Frery et al. (2014) got the second-best result. The result could, however, be improved if better text representations could be discovered, as Frery et al. (2014) claim.

2.3. Differential features based on random forest (2015)

The 2015 task is cross-genre and cross-topic authorship verification between a set of documents whose author is A and an unknown document whose author claims to be A. Bartoli et al. (2015) aggregated outputs of all random forest regressors for different feature groups, i.e. word grams, character grams, part-of-speech tags, etc. to create a single feature vector for each verification problem. Another random forest regressor is then applied to the derived feature vector to perform verification. Bartoli et al. (2015) received varied results on different language inputs. The best result was achieved for Spanish, but uncompetitive outcomes were obtained for English, Dutch and Greek.

2.4. Deep Bayes factor scoring (2020)

The 2020 task is to identify whether a pair of texts are written by the same person in a cross-domain context. Boenninghoff et al. (2020) introduced a hybrid approach, sequentially combining a Siamese network to perform feature extraction and a probabilistic model to perform Bayes factor scoring. Before that, texts are pre-processed in three steps to deal with topic influences. Firstly, rare characters and tokens are replaced with a special token. Secondly, the tokenisation process is incorporated with adding a contextual prefix and a sliding window to avoid strict sentence boundary. Thirdly, pairs of similar and different authors are resampled to increase the size of the training dataset in each training epoch. Boenninghoff et al. (2020) achieved outstanding results, with an AUC (area under ROC curve) score of 0.969 and an F1 score of 0.936 in the large training dataset with text inputs of 21,000 characters. Boenninghoff et al. (2020) got the best result in the 2020 PAN's shared task. However, one drawback of this method was the training time, which took up to 6 hours.

2.5. Key takeaways

Generally, all four approaches have complex feature selections and multiple pre-processing steps. Inspired by the Occam's razor, this study introduces ChunkedHCs, offering a simpler work procedure than the machine/deep learning approaches. The following sections will discuss the ChunkedHCs' foundations, details of the algorithm and its promising verification ability.

3. Theoretical foundations

This section will introduce two fundamental theoretical foundations, which are the core components of ChunkedHCs. They include statistical testing Higher Criticism (Section 3.1) and the HC-based similarity algorithm (Section 3.2).

3.1. Higher Criticism (HC)

3.1.1. HC testing

Donoho and Jin (2004) proposed a statistical testing Higher Criticism (HC) to test a very subtle problem whether, from a large number of independent statistical tests, all of the null hypotheses are true (global null hypothesis) or some of them are not (global alternative hypothesis). Suppose there is a set of N p-values π_i, where $\pi_i \in \{\pi_1, ..., \pi_N\}$ is the sorted p-values from π_1 (smallest) to π_N (biggest). The HC objective function $HC(i, \pi_i)$ is defined as

$$HC(i, \pi_i) = \sqrt{N}\ \frac{i/N - \pi_i}{\sqrt{i/N(1 - i/N)}}\ ;\ i = 1, ..., N. \tag{1}$$

The HC statistic HC^* is then formularised as

$$HC^* = \max_{1 \le i \le \alpha N} HC(i, \pi_i), \tag{2}$$

where $\alpha \in (0, 1]$ is the tuning parameter. The objective function $HC(i, \pi_i)$, which can be considered as the "Z-score of the p-value", reveals an asymptotically normal distribution $N(0, 1)$ under the global null hypothesis. For any π_i, the goal is to seek the largest standardised discrepancy, i.e. the HC statistic HC^*, between the observed behaviour and the expected behaviour under the global null hypothesis. When HC^* is large, the set of p-values is inconsistent with the global null hypothesis. In other words, it is more likely to have enough evidence to support the global alternative hypothesis that there is a presence of the non-null hypotheses (Donoho and Jin, 2008). The computational complexity of calculating the HC statistic is $O(Nlog(N))$, which is considered as moderate (Donoho and Jin, 2015).

3.1.2. Feature selection from HC threshold

From (1) and (2), the index i^* is identified where $HC(i, \pi_i)$ is maximum, such that

$$i^* \leftarrow \max_{1 \le i \le \alpha N} HC(i, \pi_i). \tag{3}$$

Correspondingly, the HC threshold t_{HC}, which is the p-value at $(i^*)^{th}$ index, is defined as

$$t_{HC} = \pi_{i^*}. \tag{4}$$

Any p-values which are smaller than the threshold t_{HC} are responsible for the magnitude of the HC statistic HC^*. Furthermore, since the set of N p-values $\pi_i \in \{\pi_1, ..., \pi_N\}$ is ascendingly sorted, the set of p-values influencing HC^* is thus $\{\pi_1, ..., \pi_{i^*}\}$. The indexes $1, ..., i^*$ associated with the set $\{\pi_1, ..., \pi_{i^*}\}$ are useful for identifying important features in classification settings (Donoho and Jin, 2009).

3.2. HC-based similarity algorithm

Kipnis (2020a & 2020b) introduced a binomial word allocation model, consequently using the statistical testing HC as a similarity measure between two documents. It is called the HC-based similarity algorithm. This section explains the algorithm step-by-step.

3.2.1. Binomial word allocation model

Given two documents D_1 and D_2 with a vocabulary $\mathbb{W}$, there is a word $w \in \mathbb{W}$ which might occur in either D_1 or D_2, or both. The number of times the word w occurs in D_1 and D_2 are $N(w|D_1)$ and $N(w|D_2)$ respectively. The word-frequency tables associated with D_1 and D_2 are $\{N(w|D_1), w \in \mathbb{W}\}$ and $\{N(w|D_2), w \in \mathbb{W}\}$ respectively.

Now, considering a specific word w, the number of the word w occurring in D_1 is

$$x_w = N(w|D_1). \tag{5}$$

The number of times the word w occurs in D_1 and D_2 is

$$n_w = N(w|D_1) + N(w|D_2). \tag{6}$$

The probability of obtaining the word w in document D_1 is

$$p_w = (|D_1| - x_w) / (|D_1| + |D_2| - n_w), \tag{7}$$

where $|D_1|$ and $|D_2|$ are the sizes of D_1 and D_2. Kipnis (2020a) proposed a binomial word allocation model with the null hypothesis H_o that the word w symmetrically occurs in both D_1 and D_2, in other words, following the binomial distribution $Bin(n_w, p_w)$ across the two documents equally. Regarding the binomial distribution, n_w is the number of trials, and p_w is the probability of success on a single trial. The hypothesis test is

$$\begin{aligned} H_o &: N(w|D_1) \sim Bin(n_w, p_w), \\ H_a &: N(w|D_1) \sim Bin(n_w, p_w). \end{aligned} \tag{8}$$

Accordingly, the p-value derived from x_w, n_w and p_w for the word w is

$$\pi(w|D_1, D_2) = Prob(|Bin(n_w, p_w) - n_w p_w| \ge |N(w|D_1) - n_w p_w|). \tag{9}$$

Consequently, the set of p-values $\{\pi(w|D_1, D_2)\}_{w \in \mathbb{W}}$ can be obtained by performing multiple tests for all words in the

vocabulary $\mathbb{W}$. This set of p-values will be used in the next step for calculating HC statistics.

3.2.2. HC statistics for measuring similarity

As mentioned in Section 3.1.1, the statistical testing HC is used to tell, from a set of p-values, whether there exists non-null hypotheses among many other null hypotheses from the magnitude of the HC statistic HC^*.

In the context of the binomial word allocation model, HC^* is now derived from the set of the ascendingly sorted p-values $\pi_i \in Sort(\{\pi(w|D_1, D_2)\}_{w\in\mathbb{W}})$. HC^* is used to detect whether any words w in the vocabulary $\mathbb{W}$ does not symmetrically occur in both D_1 and D_2. When HC^* is large, there are words w whose occurrences are symmetrically different in D_1 and D_2. To a certain extent, it indicates the writing styles in D_1 and D_2 are different from each other at a word level. Kipnis (2020a) uses HC^* as the measure of distance $d(D_1, D_2)$ between D_1 and D_2. The smaller $d(D_1, D_2)$ is, the more likely D_1 and D_2 were written by the same author; and vice versa.

$$HC^*,\ i^* \leftarrow \max_{1\le i\le \alpha N} HC(i, \pi_i) \tag{10}$$

$$d(D_1, D_2) = HC^*. \tag{11}$$

3.2.3. Identifying distinguishing words between two documents

As explained in Section 3.1.2, HC is also used for selecting important features from the HC threshold. For the HC-based similarity algorithm, the threshold is π_{i^*}, where the index i^* is derived from (10). As the subset of $\mathbb{W}$, the set of distinguishing words Δ between D_1 and D_2 will be

$$\Delta \leftarrow \{w \in \mathbb{W} :\ \pi(w|D_1, D_2) \le \pi_{i^*}\};\ \Delta \subseteq \mathbb{W} \tag{12}$$

From the empirical results of Kipnis (2020a), words in Δ from homogeneous authorship have small p-values and small variance. Those words, which consistently occur across documents with different topics, are considered as author-characteristic words. Furthermore, words in Δ directly influence the statistical testing HC or the distance $d(D_1, D_2)$ between two documents. By contrast, words which have large variance tend to occur more frequently in a specific topic but not frequently in other topics. Such words, which are considered as topic-related words, do not affect $d(D_1, D_2)$. This is an important property of the HC-based similarity algorithm, which automatically focuses on author-characteristic words rather than topic-related words. The algorithm is thus robust and suitable to deal with authorship tasks in different topics.

3.2.4. Summary

Using the HC-based similarity algorithm for training the model is straightforward and fast. To construct word-frequency tables, the text input is only pre-processed by removing names, numbers and/or punctuations. Then, specifying the vocabulary $\mathbb{W}$ is done by choosing $N = |\mathbb{W}|$ number of the most frequently used words. Regarding the output, sets of distinguishing words Δ produced by individual authors and the distributions of those words are clearly identified. This makes the output highly interpretable, since comparing such unique patterns of Δ is visually possible. However, without knowing how large the statistic HC should be, it is challenging to identify whether two documents are actually sampled from the same author. Therefore, ChunkedHCs could offer more confident results by providing a similarity probability from a pair of texts.

Algorithm 1. HC-based Similarity

Algorithm 1. HC-based Similarity

Input: Two word-frequency tables $\{N(w|D_1), w \in \mathbb{W}\}$ and $\{N(w|D_2), w \in \mathbb{W}\}$

Step 1: Performing multiple binomial tests

$|D_1| \leftarrow \sum_{w\in\mathbb{W}} N(w|D_1)$

$|D_2| \leftarrow \sum_{w\in\mathbb{W}} N(w|D_2)$

For each $w \in \mathbb{W}$:

$x_w \leftarrow N(w|D_1)$

$n_w \leftarrow N(w|D_1) + N(w|D_2)$

$p_w \leftarrow (|D_1| - x_w)/(|D_1| + |D_2| - n_w)$

$\pi(w|D_1, D_2) \leftarrow$ exact binomial test (x_w, n_w, p_w)

Step 2: Calculating Higher Criticism Statistic

$N \leftarrow |\mathbb{W}|$

$\pi_i \in \{\pi_1, \ldots, \pi_N\} \leftarrow Sort(\{\pi(w|D_1, D_2)\}_{w\in\mathbb{W}})$

$HC(i, \pi_i) \leftarrow \sqrt{N}(i/N - \pi_i)\big(i/N(1 - i/N)\big)^{-1/2}$

$HC^*, i^* \leftarrow \max_{1\le i\le \alpha N} HC(i, \pi_i)$

$d(D_1, D_2) \leftarrow HC^*$

$\Delta \leftarrow \{w \in \mathbb{W}: \pi(w|D_1, D_2) \le \pi_{i^*}\}$

Output: $d(D_1, D_2), \Delta$

4. ChunkedHCs algorithm

This section introduces the ChunkedHCs algorithm used for authorship verification. The task is to identify whether a pair of texts are written by the same person. ChunkedHCs has three steps. Step 1 is chunking the texts. Step 2 is measuring HC distances between chunks and corpora. Step 3 is estimating similarity probability between the two texts.

4.1. Chunking the texts (Step 1)

Given a pair of texts, Text 1 and Text 2, each of them is split into chunks of identical lengths of characters. Text 1 is split into a set of chunks $C_1 = \{T_{11}, T_{12}, ..., T_{1i}\}$, where C_1 is the Text 1's corpus.

$$\text{Text 1} \xrightarrow{split} C_1 = \{T_{11}, T_{12}, ..., T_{1i}\}; \tag{13}$$

Similarly, Text 2 is also turned into a set of chunked texts $C_2 = \{T_{21}, T_{22}, ..., T_{2j}\}$, where C_2 is the corpus of Text 2.

$$\text{Text 2} \xrightarrow{split} C_2 = \{T_{21}, T_{22}, ..., T_{2j}\}; \tag{14}$$

The number of the chunked texts for corpus C_1 and C_2 are i and j respectively, i.e. $|C_1| = i$ and $|C_2| = j$. The sizes of all chunked texts are $\mathbb{C}$, i.e. $|T| = \mathbb{C}$. The chunk size $\mathbb{C}$ is smaller than the lengths of either Text 1 or Text 2. The text residuals, whose lengths are less than the chunk size $\mathbb{C}$, will not be used.

4.2. HC distances between chunks and corpora (Step 2)

First, a vocabulary size $\mathbb{V}$ is chosen for calculating HC distances in this step. This vocabulary size $\mathbb{V}$ is the vocabulary size $|\mathbb{W}|$ in the HC-based similarity algorithm (Section 3.2). $\mathbb{V}$ is the number of most frequently-used monograms, bigrams and trigrams in a pair of a chunk and a corpus as inspired by the approach of Kipnis (2020b). In this paper, $\mathbb{V}$ is written as the number of most frequently-used words for short.

Considering a single chunked text $T_k \in (C_1 \cup C_2)$, where $k \in [1,..., i+j]$, the HC distances between T_k with the two corpora are calculated. In that regard, d_{1k} is the HC distance between T_k with C_1, and d_{2k} is the HC distance between T_k with C_2.

To calculate d_{1k}, if T_k does not belong to C_1, d_{1k} is the HC distance between T_k and C_1. However, if T_k belongs to C_1, d_{1k} is the HC distance between T_k and C_1^*, where C_1^* is the corpus C_1 without T_k, i.e. $C_1^* = C_1 \backslash \{T_k\}$ if $T_k \in C_1$. This is known as the "Leave-one-out" method. The same principle is applied for calculating d_{2k}. Therefore, the distances are calculated as

$$d_{1k} = \begin{cases} d_{HC}(T_k, C_1);\ T_k \notin C_1 \\ d_{HC}(T_k, C_1^*);\ T_k \in C_1 \end{cases} \tag{15}$$

$$d_{2k} = \begin{cases} d_{HC}(T_k, C_2);\ T_k \notin C_2 \\ d_{HC}(T_k, C_2^*);\ T_k \in C_2 \end{cases} \tag{16}$$

From the HC-based Similarity Algorithm, the smaller the HC distance is, the more likely texts are sampled from the same author. Thus, if d_{1k} is smaller than d_{2k}, T_k will be grouped into the set of the predicted Text 1's chunks C_{1pred}. Likewise, if d_{1k} is greater than d_{2k}, T_k will be grouped into the set of the predicted Text 2's chunks C_{2pred}.

$$\begin{cases} d_{1k} < d_{2k} \Rightarrow C_{1pred} \leftarrow T_k, \\ d_{1k} > d_{2k} \Rightarrow C_{2pred} \leftarrow T_k. \end{cases} \tag{17}$$

Next, equations (15)–(17) are applied for all chunked texts $T_k \in (C_1 \cup C_2)$. After performing those calculations, the total number of all predicted chunked texts is equal to the total number of all chunked texts, i.e.

$$\left|C_{1pred}\right| + \left|C_{2pred}\right| = |C_1| + |C_2|. \tag{18}$$

4.3. Estimating similarity probability (Step 3)

Based on C_1, C_2, C_{1pred} and C_{2pred}, the classification of the chunked texts is evaluated. Table 1 illustrates the confusion matrix of the binary classification between Text 1's chunks and Text 2's chunks.

Subsequently, the Accuracy score and F1 values are calculated. These classification metrics are then used to calculate the score

$$S = 0.5 \times (Accuracy + F1);\ 0 \leq S \leq 1. \tag{19}$$

If Text 1 and Text 2 are sampled from different authors and individual chunked texts are all correctly classified, the Accuracy score and F1 scores will both be 1. The desired score S for this scenario will be

$$S_{de.\ diff.} = 0.5 \times (1+1) = 1. \tag{20}$$

If Text 1 and Text 2 are sampled from the same author and all chunked texts are totally mixed up, the Accuracy score and F1 scores will both be 0.5. The desired score S for this situation will be

$$S_{de.simi.} = 0.5 \times (0.5+0.5) = 0.5. \tag{21}$$

In other circumstances, S could range from 0 to 1. To find the similarity probability for a pair of texts, ChunkedHCs has an underlying assumption that the modes of the S distributions are $S_{de.diff.}$ for a pair of two different authors and $S_{de.simi.}$ for a pair of the same author. There is no requirement for having specific values capturing the score S distribution variability, such as standard deviation, inter-quartile range or coefficient of variation. Fig. 1 is a simulation demonstrating the expected S distributions for pairs of similar authors and pairs of different authors. For pairs of similar authors, the S distribution exhibits a peak (corresponding to the mode) at $S_{de.simi.}$. Similarly, regarding pairs of different authors, the peak/mode of the S distribution occurs at $S_{de.diff.}$.

The range of S between those modes is critical to identify the optimal threshold distinguishing pairs of similar authors from pairs of different authors. Thus, values of S between 0.5 and 1 are rescaled into a new range between 0 and 1. All values of S below 0.5 are converted to 0. These transformations create the probability P estimating the likelihood of having a pair of similar authors, when P is close to 0. Or, it is likely to be a pair of different authors when P is close to 1. Fig. 2 shows the probability distribution$P$ being derived from the score S.

$$P = \begin{cases} 0 & ;\ S < 0.5 \\ \dfrac{S - S_{de.\ simi.}}{S_{de.\ diff.} - S_{de.\ simi.}} & ;\ S \geq 0.5 \end{cases} \tag{22}$$

The objective of ChunkedHCs is to identify whether a pair of texts are written by the same person. Therefore, pairs of similar authors are assigned as the positive class; meanwhile, pairs of

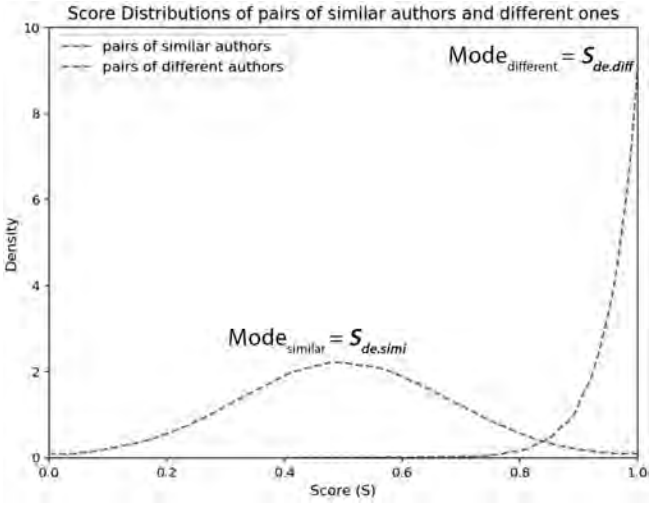

Fig. 1. The expected score S distributions.

Table 1
Confusion matrix.

	Actual Text 1's chunks	Actual Text 2's chunks	Total
Predicted Text 1's chunks	True Text 1's chunks	False Text 1's chunks	$\left\|C_{1pred}\right\|$
Predicted Text 2's chunks	False Text 2's chunks	True Text 2's chunks	$\left\|C_{2pred}\right\|$
Total	$\|C_1\|$	$\|C_2\|$	

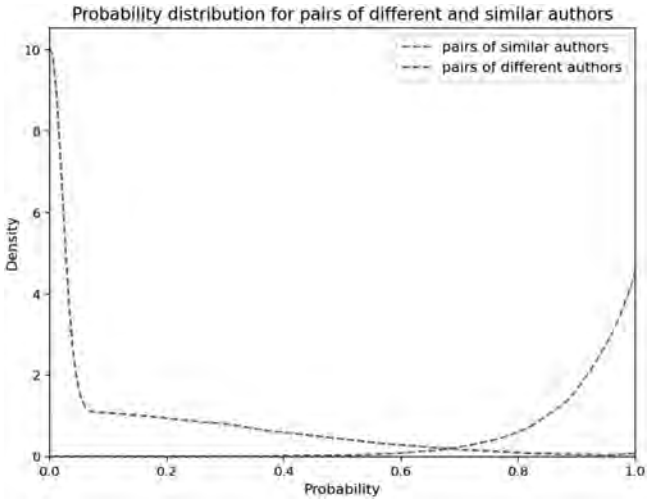

Fig. 2. P distributions transformed from S

different authors are nominated as the negative class. The similarity probability P_{simi} is accordingly derived from P (23). It is likely to be a pair of similar authors when P_{simi} is close to 1. Or, it tends to be a pair of different authors when P_{simi} is close to 0. The similarity probability distribution P_{simi} is shown in Fig. 3.

$$P_{simi} = 1 - P. \tag{23}$$

Taking (19), (20), (21), (22) and (23) all together, P_{simi} is derived from S as

$$P_{simi} = \begin{cases} 1 & ; S < 0.5, \\ 2(1-S); & S \geq 0.5. \end{cases} \tag{24}$$

4.4. Summary

Regarding the input, ChunkedHCs only require simple pre-processing methods for text pairs, such as removing names, numbers and/or punctuations. Within the scope of this study, there is no clear evidence that implementing such pre-processing steps would significantly improve the algorithm's performance. One possible explanation is that ChunkedHCs only care about the set of the distinguishing words Δ from a pre-defined vocabulary $\mathbb{V}$ (Section 3.2.3). However, further research is needed to investigate effective pre-processing procedures.

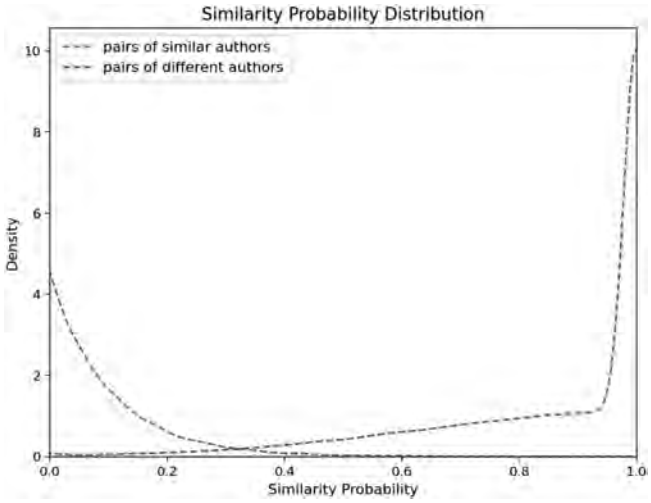

Fig. 3. P_{simi} distributions transformed from P

Considering the output, with a similarity probability P_{simi} ranging from 0 to 1, the task now is to find an optimal decision threshold τ that distinguishes if a pair of texts is sampled from the same author. The value τ and the distribution of P_{simi} would vary greatly according to the lengths of the text inputs, along with the choices of $\mathbb{C}$ and $\mathbb{V}$. Using empirical data, the next section will discuss the ChunkedHCs performance and the optimal choices of $\mathbb{C}$ and $\mathbb{V}$ for different lengths of the text inputs.

Algorithm 2. ChunkedHCs

Algorithm 2. ChunkedHCs

Input: Text 1 & Text 2
Step 1: Chunk the texts
Choosing chunk size $\mathbb{C}$

$$\text{Text 1} \xrightarrow{split} C_1 = \{T_{11}, T_{12}, \ldots, T_{1i}\};\ |C_1| = i;\ |T| = \mathbb{C}$$

$$\text{Text 2} \xrightarrow{split} C_2 = \{T_{21}, T_{22}, \ldots, T_{2j}\};\ |C_2| = j;\ |T| = \mathbb{C}$$

Step 2: Measure HC distances between chunks and corpora
Choosing vocabulary size $\mathbb{V}$ for calculating d_{HC}

$$T_k \in (C_1 \cup C_2) \longrightarrow T_k \begin{cases} d_{k1} = \begin{cases} d_{HC}(T_k, C_1);\ T_k \notin C_1 \\ d_{HC}(T_k, C_1^*);\ T_k \in C_1 \end{cases}; \\ d_{k2} = \begin{cases} d_{HC}(T_k, C_2);\ T_k \notin C_2 \\ d_{HC}(T_k, C_2^*);\ T_k \in C_2 \end{cases} \end{cases}$$

$$\longrightarrow \begin{cases} d_{k1} < d_{k2} \Rightarrow C_{1pred} \leftarrow T_k \\ d_{k1} > d_{k2} \Rightarrow C_{2pred} \leftarrow T_k \end{cases}$$

where $C_1^* = C_1 \setminus \{T_k\};\ T_k \in C_1$ & $C_2^* = C_2 \setminus \{T_k\};\ T_k \in C_2$
Step 3: Estimate Similarity Probability
From C_1, C_2, C_{1pred} & C_{2pred}:

$$S = \frac{Accuracy\ +\ F1}{2};\ \Rightarrow P_{simi} = \begin{cases} 1 & ; S < 0.5 \\ 2(1-S); & S \geq 0.5 \end{cases}$$

Output:
$P_{simi} \longrightarrow 1 \Rightarrow$ Text 1 and Text 2 are from the same author.
$P_{simi} \longrightarrow 0 \Rightarrow$ Text 1 and Text 2 are from different authors.

5. Experiments

5.1. Creating datasets from reddit

5.1.1. Why Reddit?

As one of the world's largest social media platforms, Reddit provides an abundance of users' data that could be ideally used for examining authorship verification algorithms. In October 2020, there were 430 million active users on Reddit, ranked the 13th most popular social networks worldwide (Statista, 2020). Reddit also features a wide range of topics categorised into Subreddits, e.g. Ask Reddit, Gaming, Movies, Politics, etc. This characteristic enables authorship verification algorithms to either focus on specific topics or to examine the Reddit platform as a whole. ChunkedHCs work toward the latter approach to test and demonstrate its ability for cross-topic and cross-genre verification.

5.1.2. Creating pairs of text inputs

In this study, each username is associated with his/her concatenated posts and comments on Reddit, without considering which Subreddit the individual posts and comments come from. To create a pair of texts from two different authors, texts from two

usernames are used. This approach holds an assumption that those two usernames are actually two different people. To create a pair of texts from the same author, texts from one username are split into two halves. The first half and the second half are then considered two texts in a pair of the same author. No username is used more than once. In each pair, the two texts must have the same length interval. For example, for a pair of texts in an interval between 20,000 and 30,000 characters, the first text can have 21,023 characters, while the second text could have 29,349 characters. For example, in this case, texts of 19,542 and 23,643 would not be allowed.

5.1.3. Datasets for surveying and testing ChunkedHCs

The public directory, https://files.pushshift.io/reddit/comments/, was used to create three datasets which include two surveying datasets and one testing dataset (see Table 2). Short texts considered as having less than 10,000 characters; meanwhile, texts with more 10,000 characters are referred to long texts. For each interval, the number of pairs of similar and different authors are equal.

Two datasets used for surveying ChunkedHCs (Section 5.2) were created from the Reddit users' data during June 2012. The first surveying dataset contains 1800 pairs of short texts having intervals from 1000−2000 to 9000−10,000 characters. The second surveying dataset consists of 1800 pairs of long texts, whose intervals are from 10,000−20,000, to 90,000−100,000 characters. For each of the surveying datasets, there are 1800 unique authors for 900 pairs of different authors and 900 unique authors for 900 pairs of similar authors.

There is one dataset for testing the algorithm (Section 5.3), which was obtained from Reddit users' data during December 2012. This testing dataset has 11,800 pairs of short and long texts with intervals from 1000−2000 to 29,000−30,000 characters. There are 11,600 unique authors for 5800 pairs of different authors and 5800 unique authors for 5800 pairs of similar authors.

5.2. Surveying ChunkedHCs

5.2.1. Motivations

In this section, all of the text pairs are used to survey the algorithm to find the ideal combinations of chunk size $\mathbb{C}$ and vocabulary size $\mathbb{V}$ with respect to different input lengths. The choices of $\mathbb{C}$should be large enough to sufficiently represent the authors' writing styles in individual chunks. Moreover, $\mathbb{C}$ should also be short enough to have a sufficient number of chunks for reliable classification results. In terms of $\mathbb{V}$, there is no comprehensive guide at this stage, but experiments are conducted with a variety of choices.

AUC scores are used to evaluate ChunkedHCs. The justification of using AUC is that it reveals the classification power differentiating the positive class (pairs of similar authors) from the negative class (pairs of different authors), without considering decision thresholds τ or the predicted classes. AUC only cares about the actual classes and the similarity probability P_{simi}, which is directly derived from ChunkedHCs.

In short, the purpose of this section is to provide a broad overview of how ChunkedHCs performs under different settings. Meanwhile, Section 5.3 will look more closely at the algorithm's ability for the authorship verification task with specific choices of $\mathbb{C}$ and $\mathbb{V}$, for a mix of long and short texts.

5.2.2. Short text inputs (1000−10,000 characters)

For short texts between 1000 and 10,000 characters, $\mathbb{C}\in$ {100, 200, 300, 500, 800, 1,000, 2,000, 3000} and $\mathbb{V} \in$ {10, 20, 30, 50, 80, 100, 200, 300} are used. Fig. 4 demonstrates the heatmap of the AUC scores with different combinations of $\mathbb{C}$ and $\mathbb{V}$ for the entire short-texts dataset. ChunkedHCs has poor performance for $\mathbb{C}$ values of 100 or 200 characters, when it is only slightly better than random guesses (i.e. AUC scores just above 0.5). Meanwhile, the largest chunk size at 3000 characters offers worse performance than other smaller chunk sizes. Interestingly, combining $\mathbb{C}$ of 500 characters with $\mathbb{V}$ of 20 most frequently-used words provides the best results for the short texts. Furthermore, $\mathbb{C} = 2000$ characters with $\mathbb{V}\in$ [50, 300] most frequently-used words show reasonable AUC scores.

Fig. 5 demonstrates AUC scores for all length intervals from ten combinations of $\mathbb{C}$ and $\mathbb{V}$ in which ChunkedHCs has the highest AUC scores for entire short-text dataset. This indicates that short chunk sizes at 500 and 800 characters could be suitable choices for texts from 1000 to 4000 characters. For texts between 4000 and 6000 characters, there are mixed signals among the nominated chunk sizes. For texts between 6000 and 10,000 characters, $\mathbb{C} =$ 2000 offers good results in all cases shown.

5.2.3. Long text inputs (10,000−100,000 characters)

For long texts between 10,000 and 100,000 characters, values of $\mathbb{C}\in$ {500, 1,000, 2,000, 3,000, 5,000, 8,000, 10,000} and $\mathbb{V} \in$ {50, 100, 200, 300, 500, 800, 1000} are examined. Fig. 6 demonstrates the heatmap of the AUC scores with different values of $\mathbb{C}$ and $\mathbb{V}$ for the entire long-texts dataset. ChunkedHCs shows poor performance for the shortest $\mathbb{C}$ of 500 characters, with $\mathbb{V}$ from 200 to 1000 most frequently-used words. The performance decreases when $\mathbb{C}$ is greater than 8000 characters. The algorithm seems to have desirable performance with $\mathbb{C}$ values of 2,000, 3000 or 5000 characters.

Fig. 7 illustrates AUC scores for all length intervals from ten combinations of $\mathbb{C}$ and $\mathbb{V}$ in which ChunkedHCs has the highest AUC scores for entire long-text dataset. Firstly, longer texts correspond to higher AUC scores generally. Secondly, there are only subtle differences for AUC scores among the combinations of $\mathbb{C}$ and $\mathbb{V}$ for all intervals. In particular, such differences become smaller and smaller when the texts are longer.

5.3. Testing ChunkedHCs

5.3.1. Motivations

This section is testing ChunkedHCs by using the suggested choices of $\mathbb{C}$ and $\mathbb{V}$ from Section 5.2 for mixed-ranged text pairs, whose intervals range from between 1000−2000 characters to

Table 2
Datasets for surveying and testing ChunkedHCs

	Intervals	No. intervals	No. pairs per interval	No. pairs
Short texts (Surveying)	1k-2k, …, 9k-10k	9	200	1800
Long texts (Surveying)	10k-20k, …, 90k-100k	9	200	1800
Mixed texts (Testing)	1k-2k, …, 29k-30k	29	400	11,600

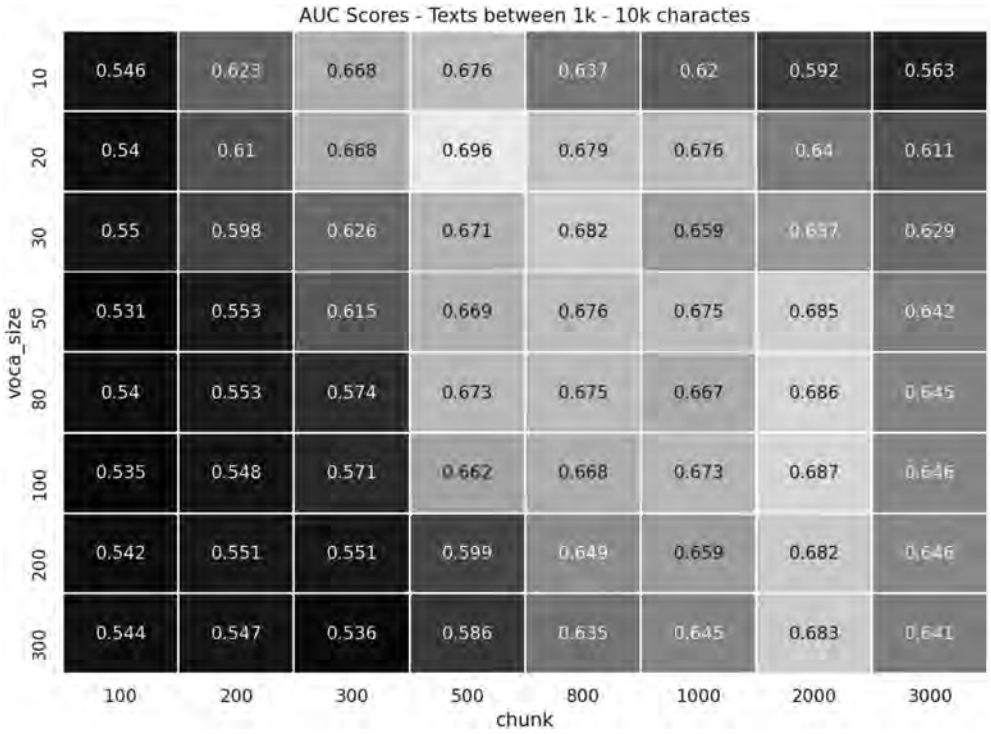

Fig. 4. Heatmap – AUC scores with different Chunk sizes and Vocabulary sizes (Short texts).

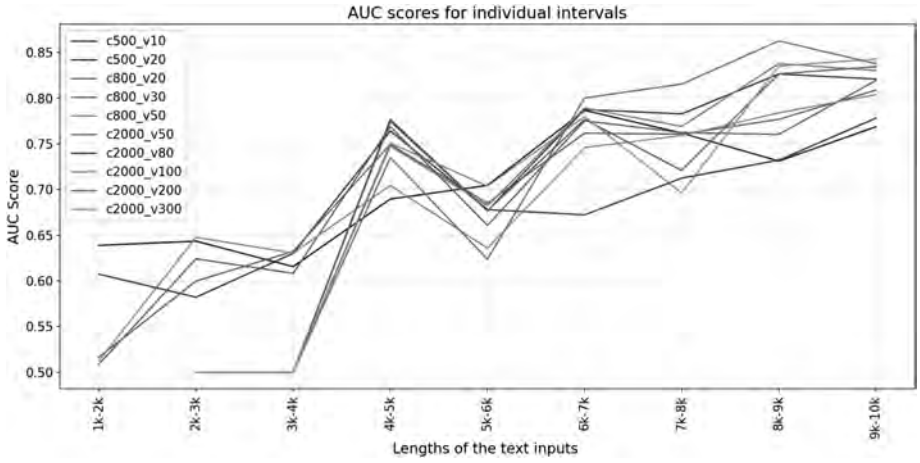

Fig. 5. Line plot – AUC scores for individual intervals (Short texts).

between 29,000–30,000 characters (see Table 3). Firstly, with the selected values of $\mathbb{C}$ and $\mathbb{V}$, the similarity probability P_{simi} is calculated for all text pairs (Step 1). Next, the dataset is split into two equal halves, which are validation set and test set. For each interval, the validation set is used for selecting optimal decision thresholds τ, where the accuracy is maximum (Step 2). Then, on the test set, τ is used to predict the positive/negative class, i.e. pairs of similar authors where P_{simi} is greater/less than τ respectively. Classification metrics including accuracy, F1 and AUC scores are calculated for individual intervals (Step 3). Overall, this section provides a detailed picture of the algorithm's ability and its limits.

5.3.2. Calculating similarity probability (Step 1)

With the chosen values of $\mathbb{C}$ and $\mathbb{V}$, P_{simi} is calculated for all text pairs. Fig. 8 demonstrates the P_{simi} distributions for pairs of both similar and different authors. Noticeably, text pairs below 10,000 characters tend to have multimodal distributions. To some degree, this might show how uncertain the algorithm is in identifying pairs of either similar or different authors in this range. By contrast, for texts above 10,000 characters, P_{simi} mostly has unimodal distributions. Furthermore, as texts get longer, the P_{simi} distributions for the two classes become more and more distinct. This indicates that ChunkedHCs might have more confident classification decisions for longer texts.

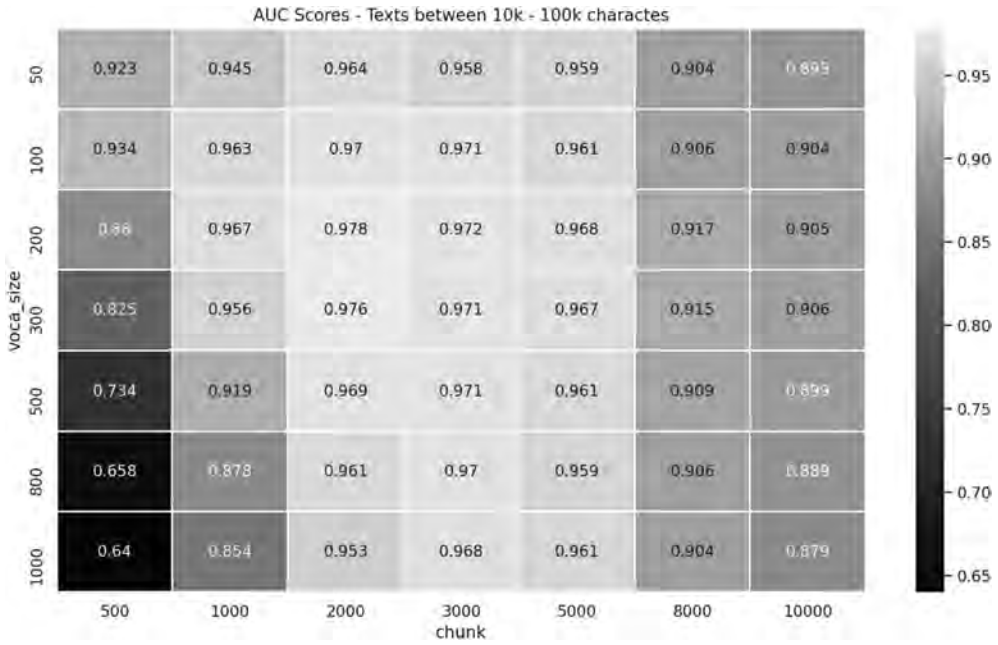

Fig. 6. Heatmap – AUC scores with different Chunk sizes and Vocabulary sizes (Long texts).

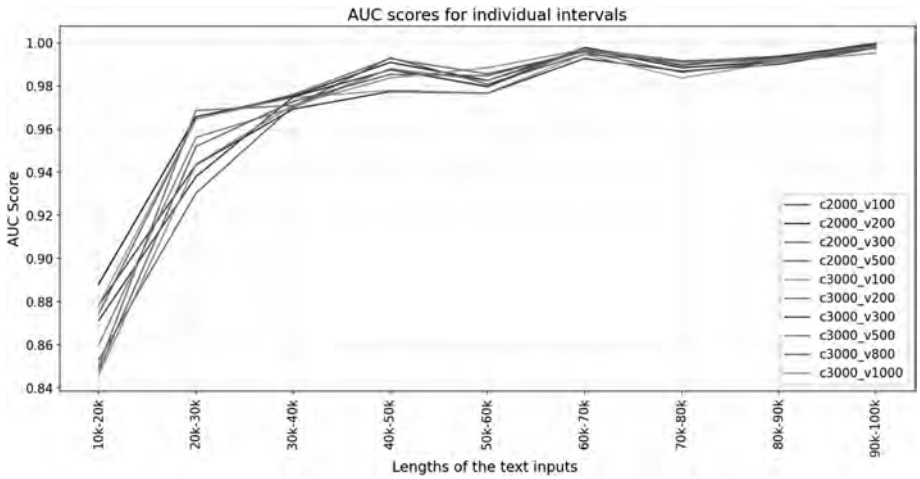

Fig. 7. Line plot – AUC scores for individual intervals (Long texts).

Table 3
Dataset for testing ChunkedHCs

Intervals	No. Intervals	Chunk size	Voca. size
1k-2k, …, 5k-6k	5	500	20
6k-7k, …, 9k-10k	4	2000	100
10k-11k, …, 29k-30k	20	2000	200

5.3.3. Choosing thresholds on the validation set (Step 2)

Within the scope of this study, accuracy is the main classification metric quantifying how well ChunkedHCs identifies pairs of similar authors. This is the basis for choosing the decision thresholds τ, where the accuracy scores are maximum on the validation set. Fig. 9 indicates how the threshold values change across the text lengths. For the intervals from 1000 to 6000 characters, τ are larger than other threshold values. For longer texts, τ seems to get approach common threshold value at 0.5.

5.3.4. Performance on the test set (Step 3)

With the optimal thresholds derived from the validation set, positive and negative classes are predicted on the test set. Table 4 shows the AUC, accuracy and F1 scores for all intervals. Firstly, for the shortest interval between 1000 and 2000 characters, the accuracy and F1 are 0.645 and 0.6785 respectively. For texts between 5000 and 10,000 characters, accuracy scores are above 0.7. For longer texts, all performance metrics increase. In particular, the accuracy and F1 reach up to 0.94 and 0.9381 respectively for the longest texts between 29,000 and 30,000 characters. This is a promising result indicating that ChunkedHCs can accurately identify if a pair of texts is written by the same person, especially when

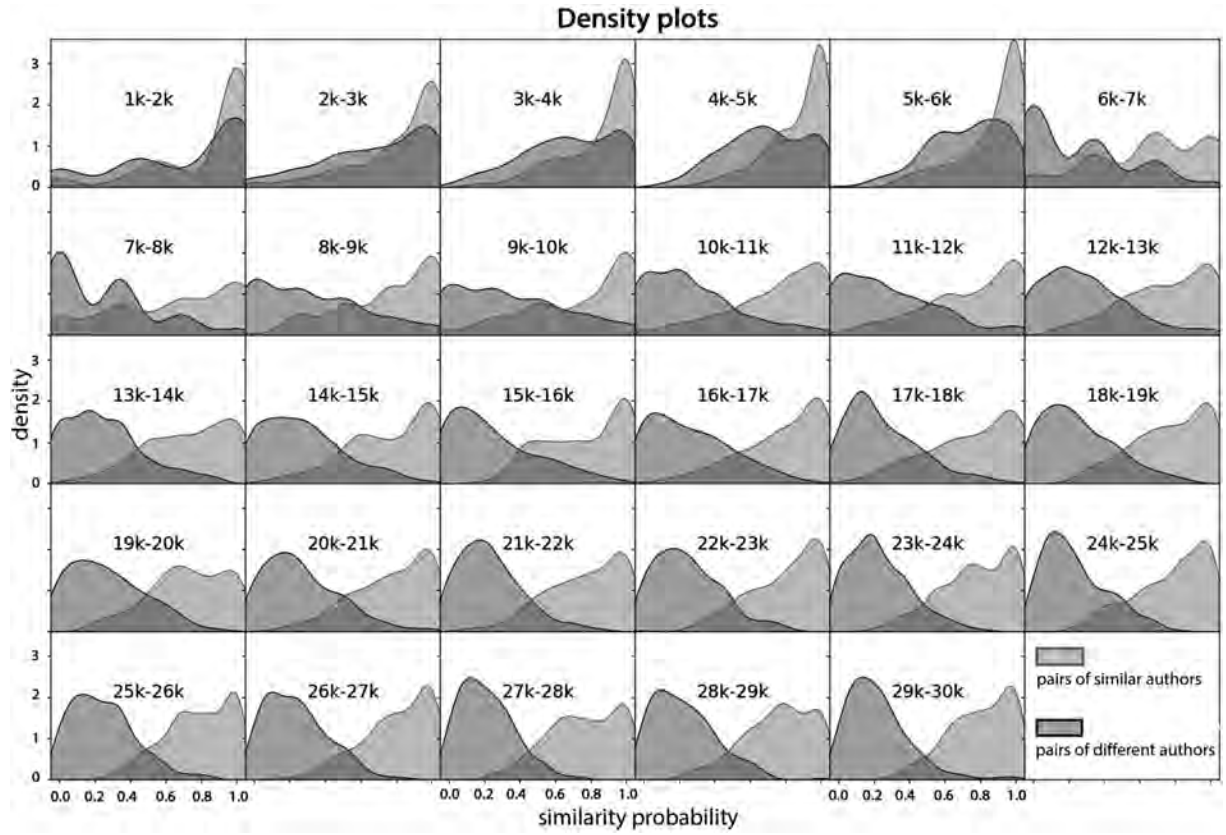

Fig. 8. Density plots — Similarity Probability Distribution.

the texts are long enough.

6. Discussion

6.1. Input

This study of ChunkedHCs uses Reddit users' data featuring informal English language that could be common among other social media platforms. Nonetheless, there could be differences regarding writing styles among those platforms. Therefore, future studies of ChunkedHCs will investigate further whether the algorithm could perform reliably on other datasets gathered from Facebook, Twitter and even dark web forums. Furthermore, the performance of ChunkedHCs on the verification task across various social media platforms will also be examined in future studies.

Inherited from the HC-based similarity algorithm, ChunkedHCs only focuses on the set of distinguishing words Δ. Such words are author-characteristic rather than topic-related (Section 3.2.3). This

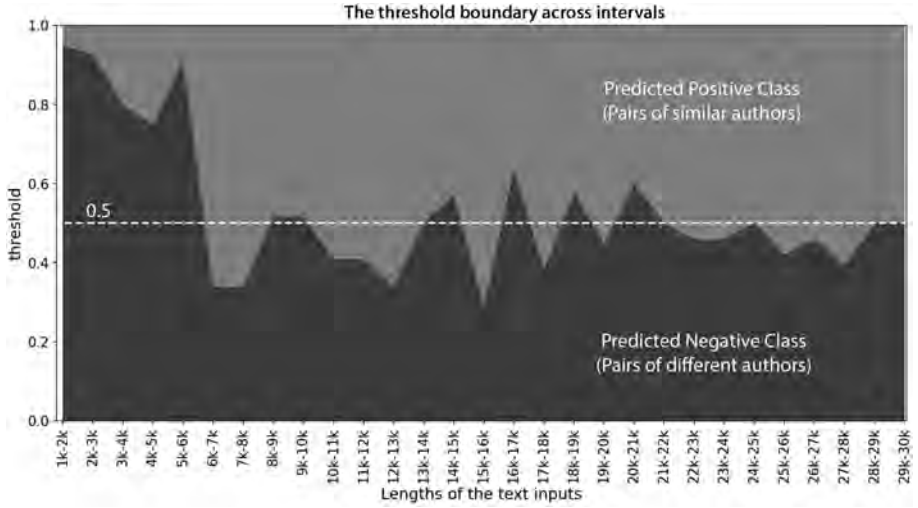

Fig. 9. Threshold boundary predicting positive and negative classes.

Table 4
AUC, Accuracy and F1 scores on the test set.

Range	AUC	Accuracy	F1	Range	AUC	Accuracy	F1	Range	AUC	Accuracy	F1
1k-2k	0.645	0.645	0.6758	**11k-12k**	0.9041	0.825	0.8259	**21k-22k**	0.9525	0.8844	0.8821
2k-3k	0.6562	0.645	0.6203	**12k-13k**	0.8762	0.815	0.8311	**22k-23k**	0.9614	0.885	0.8867
3k-4k	0.6699	0.655	0.6699	**13k-14k**	0.9211	0.8	0.7701	**23k-24k**	0.9607	0.895	0.8934
4k-5k	0.7097	0.695	0.7289	**14k-15k**	0.9044	0.81	0.8	**24k-25k**	0.959	0.8844	0.8821
5k-6k	0.7361	0.7	0.6667	**15k-16k**	0.8932	0.825	0.8472	**25k-26k**	0.9552	0.875	0.8792
6k-7k	0.8101	0.74	0.7374	**16k-17k**	0.928	0.805	0.7719	**26k-27k**	0.9732	0.8844	0.8878
7k-8k	0.7849	0.715	0.6952	**17k-18k**	0.9058	0.8241	0.8293	**27k-28k**	0.9838	0.935	0.9378
8k-9k	0.8572	0.78	0.7684	**18k-19k**	0.9255	0.803	0.7746	**28k-29k**	0.9688	0.92	0.92
9k-10k	0.8104	0.745	0.7182	**19k-20k**	0.9354	0.87	0.8725	**29k-30k**	0.9866	0.94	0.9381
10k-11k	0.8242	0.78	0.7843	**20k-21k**	0.9136	0.825	0.8128				

characteristic offers two major benefits. Firstly, the algorithm does not require sophisticated pre-processing steps. Secondly, ChunkedHCs could be theoretically suitable for other languages than English.

However, there is speculation that ChunkedHCs might not adequately capture an author's writing style due to the Zipf distribution of monograms. Frequencies of the most used monograms quickly drop towards zero such that most words' frequencies are disproportionately small. Consequently, ChunkedHCs could not sufficiently make use of the given vocabulary that only a few members in Δ derived from $\mathbb{V}$ could overinfluence the statistic HC. The underrepresented features would potentially cost the algorithm the verification ability. This suggests ChunkedHCs could be further improved by having better word representations so that signals from rarer words could be well detected.

This study of ChunkedHCs used datasets featuring balanced classes, in which the number of text pairs of similar authors is equal to the number of text pairs of different authors. This situation would not be true in reality and a class imbalance is likely to occur. It is advisable that the class imbalance should be cautiously considered when applying ChunkedHCs. It is suggested that resampling the datasets, choosing appropriate classification metrics and then accordingly adjusting the decision thresholds τ could be one possible solution responding to the class imbalance. This will be investigated in future studies.

It should be acknowledged that this study is limited to only using text pairs having the same length intervals (Section 5.1.2), as it is intended to closely examine how individual length intervals influence the ChunkedHCs performance. Therefore, the results presented might not be applicable for circumstances involving text pairs having different lengths. It is recommended that there should not be a huge difference between lengths of the two text inputs to avoid bias in the internal binary classification (Table 1).

6.2. Output

Given a pair of texts, ChunkedHCs directly provide a similarity probability. It would be reasonably confident to make decisions if the similarity probability is either 0 or 1. In most cases, it may be more challenging to conclude with less definitive probability values. Halvani et al. (2016) argued that the biggest obstacle to the authorship verification task is choosing a universal threshold distinguishing the positive/negative classes, i.e. pairs of similar/different authors. The threshold might need to be adjusted for different datasets and may also depend on other factors.

6.3. Performance

ChunkedHCs provide good results when text inputs are sufficiently long. It is crucial to carefully select chunk size, vocabulary size and decision threshold for different input lengths in order to have desirable classification outcomes. From the empirical results shown herein, optimal choices of chunk sizes, vocabulary sizes and decision thresholds are suggested for different input lengths. In essence, for short texts, the chunk size should be small and the decision threshold needs to take a more conservative value closer to the upper limit of 1. For long texts, the chunk size should be approximately 2000 to 3000 characters, with the common threshold of 0.5. However, it should be stressed that these figures are merely suggestive based on the empirical results presented. Optimisation will be required to improve the algorithm performance further.

6.4. Runtime

ChunkedHCs offer a fast and straightforward workflow. Firstly, ChunkedHCs directly provide a similarity probability from a pair of text. Without the training phase, ChunkedHCs could be considered as an unsupervised learning algorithm since there is no training-data phase. Secondly, calculating the similarity probability is relatively fast, since measuring the HC statistic has a moderate computational cost as outlined in Section 3.1.1. However, the algorithm will be slower if it has to deal with longer texts, or have shorter chunk sizes with larger vocabulary sizes.

6.5. Comparison with other verification approaches

Compared with other best intrinsic verification approaches presented in the PAN's shared tasks (see Section 2), ChunkedHCs offer a much simpler workflow. According to principles of the Occam's Razor, simpler models having fewer variables are preferable. Given different scenarios, simpler models could have better generalisation over more complex ones. In the case of ChunkedHCs, the algorithm automatically focuses on important features of the text inputs without having complicated pre-processing steps. Nevertheless, it might come at the cost of the verification ability, such that ChunkedHCs performance might be not as good as expected.

This preliminary paper serves as an introduction to ChunkedHCs; as such, the algorithm was not fully compared with other authorship verification approaches in Section 2 due to two main reasons. Firstly, the datasets used for PAN's shared tasks are not accessible for non-contestants, hence this study relied on publicly available data, as outlined in Section 5.1. Secondly, time constraints, this study could not reconstruct the PAN's approaches to be compared with ChunkedHCs on the same dataset. Future studies will comprehensively compare ChunkedHCs with other verification approaches.

7. Conclusion and future work

This paper has outlined current effective authorship verification

approaches and important theoretical foundations to introduce and examine a new algorithm, namely ChunkedHCs. ChunkedHCs is an algorithm specifically designed for the authorship verification task to identify whether a pair of texts are written by the same person. It is based on the statistical testing Higher Criticism (Donoho and Jin, 2004) and HC-based similarity algorithm (Kipnis, 2020a & 2020b) (Kestemont et al., 2020). Correspondingly, there are two key takeaways responding to the research questions outlined in the beginning as follows:

Given a pair of texts, ChunkedHCs will provide a similarity probability from 0 to 1. The lower limit indicates the pair was written by two different people; meanwhile, the upper limit suggests the same person composed the texts. The algorithm requires careful selection of chunk sizes, vocabulary sizes and decision thresholds for different input lengths. However, ChunkedHCs only require simple pre-processing steps such as removing names, numbers and/or punctuations.

Applying ChunkedHCs on the Reddit users' data indicated that the algorithm's performance improves for longer text inputs. ChunkedHCs achieve decent classification results for short texts with an accuracy of 0.645 and an F1 of 0.6758 for shortest texts between 1000 and 2000 characters. For long texts, ChunkedHCs obtain impressive outcomes with an accuracy of up to 0.94 and an F1 of 0.9381 for texts between 29,000 and 30,000 characters.

This paper is an introductory study of ChunkedHCs, which demonstrates great potential for authorship verification applications. However, it is acknowledged that this topic requires in-depth further research in several areas. Firstly, as outlined in Section 6.1, one possible solution helping ChunkedHCs have better word representations is grouping rarer words sharing some similarity together. In particular, mapping all words to their corresponding Part of Speech (POS) tags with some modifications for the text inputs have shown highly promising results as this paper's research group is currently studying. Secondly, as Section 6.3 highlighted, future research will involve optimising key parameters, including the chunk size, vocabulary size and decision threshold. In this paper, only simple grid searches are used to find the optimal values. Therefore, it is highly likely that ChunkedHCs could be further improved with more sophisticated optimisation methods. Thirdly, there will be an expansion of the ChunkedHCs study to other datasets in different languages so that the algorithm's generalisation could be examined and expectedly justified. Finally, comparative studies will also be conducted to critically evaluate ChunkedHCs performance with other authorship verification approaches.

Acknowledgement

This project is primarily inspired by the work of Kipnis (2020a & 2020b). The code implementation of this project is based upon Kipnis's public directory at https://github.com/alonkipnis/AuthorshipAttribution.

References

Bartoli, A., Dagri, A., Lorenzo, A.D., Medvet, E., Tarlao, F., 2015. An author verification approach based on differential features – notebook for PAN at CLEF 2015. In: Cappellato, L., Ferro, N., Jones, G., Juan, E.S. (Eds.), CLEF 2015 Evaluation Labs and Workshop – Working Notes Papers, 8-11 September, Toulouse. CEUR-WS.org, Frances. https://pan.webis.de/downloads/publications/papers/bartoli_2015b.pdf.

Boenninghoff, B., Rupp, J., Nickel, R.M., Kolossa, D., 2020. Deep Bayes factor scoring for authorship verification – notebook for PAN at CLEF 2020. In: Cappellato, L., Eickhoff, C., Ferro, N., Névéol, A. (Eds.), CLEF 2020 Labs and Workshops, Notebook Papers, 22-25 September, Thessaloniki, Greece. CEUR-WS.org. https://pan.webis.de/downloads/publications/papers/boenninghoff_2020.pdf.

Donoho, D., Jin, J., 2004. Higher criticism for detecting sparse heterogeneous mixtures. Ann. Stat. 32 (3), 962–994. https://arxiv.org/pdf/math/0410072.pdf.

Donoho, D., Jin, J., 2008. Higher criticism thresholding: optimal feature selection when useful features are rare and weak. Proc. Natl. Acad. Sci. Unit. States Am. 105 (39), 14790–14795. https://www.pnas.org/content/105/39/14790.

Donoho, D., Jin, J., 2009. Feature selection by higher criticism thresholding achieves the optimal phase diagram. Phil. Trans. Math. Phys. Eng. Sci. 367 (1906), 4449–4470. https://royalsocietypublishing.org/doi/10.1098/rsta.2009.0129#.

Donoho, D., Jin, J., 2015. Higher criticism for large-scale inference, especially for rare and weak effects. Stat. Sci. 30 (1), 1–25. https://projecteuclid.org/download/pdfview_1/euclid.ss/1425492437.

Facebook, 2019. Facebook Q4 2019 results [Online]. Available from. https://s21.q4cdn.com/399680738/files/doc_financials/2019/q4/Q4-2019-Earnings-Presentation-_final.pdf. (Accessed 26 January 2021).

Frery, J., Largeron, C., Juganaru-Mathieu, M., 2014. UJM at CLEF in author identification – notebook for PAN at CLEF 2014. In: Cappellato, L., Ferro, N., Halvey, M., Kraaij, W. (Eds.), Working Notes Papers of the CLEF 2014 Evaluation Labs, 15-18 September. CEUR-WS.org, Sheffield, UK. https://pan.webis.de/downloads/publications/papers/frery_2014.pdf.

Halvani, O., Steinebach, M., Zimmermann, R., 2013. Authorship verification via K-nearest neighbor estimation notebook for PAN at CLEF 2013. In: Forner, P., Navigli, R., Tufis, D. (Eds.), CLEF 2013 Evaluation Labs and Workshop – Working Notes Papers, 23-26 September. CEUR-WS.org, Valencia, Spain. https://pan.webis.de/downloads/publications/papers/halvani_2013.pdf.

Halvani, O., Winter, C., Pflug, A., 2016. Authorship verification for different languages, genres and topics. Digit. Invest. 16, S33–S43. DFRWS EU 2016. https://www.sciencedirect.com/science/article/pii/S1742287616000074.

Kestemont, M., Manjavacas, E., Markov, I., Bevendorff, J., Wiegmann, M., Stamatatos, E., Potthast, M., Stein, B., 2020. Overview of the cross-domain authorship verification task at PAN 2020. In: Cappellato, L., Eickhoff, C., Ferro, N., Névéol, A. (Eds.), CLEF 2020 Labs and Workshops, Notebook Papers, 22-25 September 2020, Thessaloniki, Greece. CEUR-WS.org. https://pan.webis.de/downloads/publications/papers/kestemont_2020.pdf.

Kipnis, A., 2020a. Higher Criticism for discriminating word-frequency tables and testing authorship. arXiv:1911.01208v3 [cs.CL]. https://arxiv.org/pdf/1911.01208.pdf.

Kipnis, A., 2020b. Higher criticism as an unsupervised authorship discriminator – notebook for PAN at CLEF 2020. In: Cappellato, L., Eickhoff, C., Ferro, N., Névéol, A. (Eds.), CLEF 2020 Labs and Workshops, Notebook Papers, 22-25 September 2020, Thessaloniki, Greece. CEUR-WS.org. https://pan.webis.de/downloads/publications/papers/kipnis_2020.pdf.

Stamatatos, S., Daelemans, W., Verhoeven, B., Potthast, M., Stein, B., Juola, P., Sanchez-Perez, M.A., Barrón-Cedeño, A., 2014. Overview of the author identification task at PAN 2014. In: Cappellato, L., Ferro, N., Halvey, M., Kraaij, W. (Eds.), Working Notes Papers of the CLEF 2014 Evaluation Labs, 15-18 September 2014. CEUR-WS.org CEUR-WS.org, Sheffield, UK. https://pan.webis.de/downloads/publications/papers/stamatatos_2014.pdf.

Statista, 2020. Most popular social networks worldwide as of October 2020, ranked by number of active users [Online]. Available from. https://www.statista.com/statistics/272014/global-social-networks-ranked-by-number-of-users/. Accessed 5th January 2021.