Who Wrote the Web? Revisiting Influential Author Identification Research Applicable to Information Retrieval

Martin Potthast,[1] Sarah Braun,[2] Tolga Buz,[3] Fabian Duffhauss,[4] Florian Friedrich,[5]
Jörg Marvin Gülzow,[6] Jakob Köhler,[7] Winfried Lötzsch,[8] Fabian Müller,[9]
Maike Elisa Müller,[3] Robert Paßmann,[10] Bernhard Reinke,[10] Lucas Rettenmeier,[5]
Thomas Rometsch,[11] Timo Sommer,[12] Michael Träger,[13] Sebastian Wilhelm,[2]
Benno Stein,[1] Efstathios Stamatatos,[14] and Matthias Hagen[1]

[1]Bauhaus-Universität Weimar, [2]Technische Universität München, [3]Technical University of Berlin, [4]RWTH Aachen University, [5]Heidelberg University, [6]University of Konstanz, [7]Free University of Berlin, [8]Chemnitz University of Technology, [9]Karlsruhe University of Applied Sciences, [10]University of Bonn, [11]University of Michigan, [12]Hamburg University of Technology, [13]University of Bamberg, and [14]University of the Aegean

martin.potthast@uni-weimar.de

Abstract In this paper, we revisit author identification research by conducting a new kind of large-scale reproducibility study: we select 15 of the most influential papers for author identification and recruit a group of students to reimplement them from scratch. Since no open source implementations have been released for the selected papers to date, our public release will have a significant impact on researchers entering the field. This way, we lay the groundwork for integrating author identification with information retrieval to eventually scale the former to the web. Furthermore, we assess the reproducibility of all reimplemented papers in detail, and conduct the first comparative evaluation of all approaches on three well-known corpora.

1 Introduction

Author identification is concerned with whether and how an author's identity can be inferred from their writing by modeling writing style. Author identification has a long history, the first known approach dating back to the 19th century [27]. Ever since, historians and linguists have tried to settle disputes over the authorship of important pieces of writing by manual authorship attribution, employing basic style markers, such as average sentence length, average word length, or hapax legomena (i.e., words that occur only once in a given context), to name only a few. It is estimated that more than 1,000 basic style markers have been proposed [31]. In the past two decades, author identification has become an active field of research for computer linguists as well, who employ machine learning on top of models that combine traditional style markers with new ones, the manual computation of which has been infeasible before. Author identification technology is evolving at a rapid pace. The field has diversified into many sub-disciplines where correlations of writing style with author traits are studied, such as age, gender, and other demographics. Moreover, in an attempt to scale their approaches, researchers apply them on increasingly large datasets with up to thousands of authors and tens of thousands of documents. Naturally, some of the document collections used for evaluation are sampled from the web, carefully ensuring that individual documents can be attributed with confidence to specific authors.

While applying this technology at web scale is still out of reach, we conjecture that it is only a matter of time until tailored information retrieval systems will index authorial style, retrieve answers to writing style-related queries as well as queries by example, and eventually, shed light on the question: Who wrote the web? Besides obvious applications in law enforcement and intelligence—a domain for which little is known about the state of the art of their author identification efforts—many other stakeholders will attempt to tap authorial style for purposes of targeted marketing, copyright enforcement, writing support, establishing trustworthiness, and of course as yet another search relevance signal. Many of these applications bring about ethical and privacy issues that need to be reconciled. Meanwhile, authorial style patterns already form a part of every text on the web that has been genuinely written by a human. At present, however, the two communities of information retrieval and author identification hardly intersect, whereas integration of technologies from both fields is necessary to scale author identification to the web.

The above observations led us to devise and carry out a novel kind of reproducibility study that has an added benefit for both research fields: we team up with a domain expert and a group of students, identify 15 influential author identification methods of the past two decades, and have each approach reimplemented by the students. By reproducing performance results from the papers' experiments, we aim at raising confidence that our implementations come close to those of the papers' authors. This paper surveys the approaches and reports on their reproducibility. The resulting source code is shared publicly. We further conduct comparative experiments among the reimplemented approaches, which has not been done before. The primary purpose of our reproducibility study is not to repeat *every* experiment reported in the selected papers, since it is unlikely that the most influential research is outright wrong. Rather, our goal is to release working implementations to both the information retrieval community as well as the author identification community, since only a few public implementations have surfaced to date. This lays the groundwork for future collaboration among both fields.

In what follows, Section 2 reviews related work and introduces the author identification papers selected, Section 3 overviews the setup of our study, Section 4 details the students' implementations and outlines reproducibility issues observed, and Section 5 reports on the first comparative evaluation of all approaches.

2 Background, Related Work, and Paper Selection

This section briefly reviews reproducibility-related research in computer science in general, and information retrieval in particular. Afterwards, we overview author identification paradigms and the papers selected for our reproducibility study.

2.1 Reproducibility in Computer Science and Information Retrieval

The reproducibility of research results that are obtained empirically determines whether the conclusions drawn from them may eventually be accepted as fact. While many of the empirical sciences have well-established best practices for reproducing research, this is not, yet, the case in the empirical branches of the comparably young field of computer science. Regardless, even sciences that have best practices currently face a reproducibility crisis: a number of studies made the news, revealing significant amounts of peer-reviewed research to be irreproducible. In the wake of these events, many computer scientists revisit their own reproducibility record and find it lacking in many respects. For brevity, we will not recite all causes for lack of reproducibility but focus on the one that relates to our contribution, namely computer science's primary research tool: software. Or rather, its absence: the vast majority of computer science research is about the development of software that solves problems of interest, but many researchers are reluctant to share their software.

Collberg et al. [9] recently assessed the availability of the pieces of software underlying 601 papers published at ACM conferences and journals; software could be collected for only 54% of the papers.[1] No attempt was made to check whether the software actually works as advertised. To identify the reasons for not sharing software, Stodden [39] conducted a survey among 134 computer scientists and found, among others, the time to clean and polish the software (77.8%), the time to deal with support questions (51.9%), a fear of supporting competing colleagues without getting credit (44.8%), and intellectual property constraints (40.0%). After all, sharing software is voluntary, and scientometrics do not yet incorporate such community services. There are counterexamples, though, such as Weka [16] and LibSVM [8], which are used across disciplines, or Terrier [28] and Blei et al.'s LDA implementation [6], which have spread throughout information retrieval (IR). Various initiatives in IR have emerged simultaneously in 2015: the ECIR has introduced a dedicated track for reproducibility [17], a corresponding workshop has been organized at SIGIR [2], and the various groups that develop Evaluation-as-a-Service platforms for shared tasks have met for the first time [19].

One of the traditional forms of reproducibility research are meta studies, where existing research on a specific problem of interest is surveyed and summarized with special emphasis on performance. For example, in information retrieval, the meta study of Armstrong et al. [3] reveals that the improvements reported in various papers of the past decade on the ad hoc search task are void, since they employ too weak baselines. Recently, Tax et al. [40] have conducted a similar study for 87 learning-to-rank papers, where they summarize for the first time which of them perform best.

Still, meta studies usually do not include a reimplementation of existing methods. Reimplementation of existing research has been conducted by Ferro and Silvello [13] and Hagen et al. [15], the former aiming for exact replicability and the latter for reproducibility (i.e., obtaining similar results under comparable circumstances). Finally, Di Buccio et al. [11] and Lin [26] both propose the development of a central repository of baseline IR systems on standard tasks (e.g., ad hoc search). They observe that even the baselines referred to in most papers may vary greatly in performance when using different parameterizations, rendering results incomparable. A parameter model, repositories of runs, and executable baselines are proposed as a remedy. When open baseline implementations are available in a given research field such as IR, this is a sensible next step, whereas in the case of author identification, there are only a few publicly available baseline implementations to date. We are the first to provide them at scale.

[1] Interestingly, Collberg et al.'s study itself has been challenged for lack of rigor and has been reproduced more thoroughly: http://cs.brown.edu/~sk/Memos/Examining-Reproducibility/

2.2 Author Identification

Authorship analysis attempts to extract information from texts based on the personal writing style of their authors. The main focus of research in this area is on *author identification* and more specifically on authorship attribution, where given a set of candidate authors and some samples of their writing, a text of unknown or disputed authorship is attributed to one of them [20, 36]. This can be viewed as either a closed-set classification task (i.e., realistic in most forensic cases where police investigations can define a small set of suspects) or an open-set classification task (i.e., realistic in web-based applications) [25]. An important variation of this task is *authorship verification* where the set of candidate authors is a singleton [42, 38]. This can be viewed as a one-class classification problem where the negative class (i.e., texts written by other authors) is huge and heterogeneous. Another dimension gaining increasing attention is *author profiling* where the task is to extract information about the characteristics of the author (e.g., age, gender, educational level, personality, etc.) rather than their identity [30].

Following the practices of text categorization, all author identification approaches comprise two basic modules: feature extraction and classification. The former is much more challenging in comparison to topic-based text classification or sentiment analysis since writing style rather than topic or sentiment has to be quantified. Unfortunately, in general, there is a lack of style-specific words. The line of research dealing with the quantification of writing style is known as *stylometry*, it has a long history [27], and plenty of measures have been proposed so far [18]. These stylometric measures fall into the following categories [36]: *lexical* (e.g., word or sentence length distribution, vocabulary richness measures, function word frequencies), *character* (e.g., character type and character n-gram frequencies), *syntactic* (e.g., POS n-gram frequencies and rewrite rule frequencies), *semantic* (e.g., semantic relationship frequencies and semantic function frequencies), and *application-dependent* features (e.g., use of greetings in email messages or font size and color in HTML documents). Low-level features like function words and character n-grams have been reported to be the most effective while higher-level features related to syntactic parse trees or semantic information are useful complements [36]. The combination of measures from different categories can enhance the performance of authorship attribution approaches [10, 43].

With respect to the classification methods, there are two main paradigms [36]: the *profile-based* approaches are author-centric and attempt to capture the cumulative style of the author by concatenating all available samples by that author and then extracting a single representation vector. Usually, generative models (e.g., naive Bayes) are used in profile-based approaches. On the other hand, *instance-based* methods are document-centric and attempt to capture the style of each text sample separately. In case only a single long document exists for one candidate author (e.g., a book), it is split into samples and each sample is represented separately. Usually, discriminative models (e.g., SVM) are exploited in instance-based approaches.

In order to reproduce a set of author identification approaches, we compiled an initial list of 30 influential papers published in the past two decades and meant to cover the main paradigms and approaches described above. Some well-known papers from the authorship attribution literature had to be excluded since their methods are based on NLP tools that are not publicly available making their reproduction infeasible within our study setup [37, 14]. Finally, since the number of students participating in this study was

Table 1. Overview of papers selected for reimplementation. Tasks include closed-set attribution (cA), open-set attribution (oA), and verification (V). Features encode character (chr), lexical (lex), or syntactical (syn) information, or mixtures (mix) thereof. The paradigms implemented are profile-based (p) and instance-based (i). Complexity of implementation ranges from easy (*) via moderate (**) to hard (***). Citations as per Google Scholar (accessed September 29, 2015).

	Publication														
	[4]	[5]	[7]	[10]	[12]	[22]	[23]	[24]	[25]	[29]	[32]	[33]	[34]	[35]	[41]
Task	cA	cA	cA	cA	cA	cA	cA	V	oA	cA	cA	cA	cA	cA	cA
Features	lex	chr	lex	mix	chr	chr	chr	lex	chr	mix	lex	syn	lex	chr	chr
Paradigm	p	i	i	i	i	p	p	i	p	p	i	i	i	p	p
Complexity	**	*	*	*	***	*	**	**	*	**	***	**	*	*	**
Citations	14	377	213	366	41	267	60	75	89	201	17	44	26	43	80
Year	09	02	02	01	11	03	03	07	11	04	12	14	06	07	03

limited, we assigned a paper to each student with the goal of maintaining the coverage of different paradigms, and, to match the complexity of a method with the student's background (computer science, mathematics, physics, engineering). The final list of selected papers alongside their basic characteristics is shown in Table 1.

Burrows' *Delta* [7] derives the deviation of function word frequencies from their norm. Keselj et al. [22] use character n-gram profiles, a method later modified for imbalanced datasets [35]. Benedetto et al. [5], Khmelev and Teahan [23], and Teahan and Harper [41] are exploiting compression models that are based on character sequences, while the approach of Peng et al. [29] can also use word sequences. These compression-based methods have also been applied to tasks like topic detection, text genre recognition, or language identification. A combination of lexical, character, and application-dependent features suitable for the e-mail domain is described by de Vel et al. [10]. Also more complicated stylometric models are among our selection. Arun et al. [4] build a graph of function words using their proximity to estimate edge weights. Escalante et al. [12] propose local histograms representing the distribution of occurrences of character n-grams within a document. Seroussi et al. [32] describe an extension of LDA topic modeling using disjoint document and author topics. Sidorov et al. [33] make use of syntactic n-grams based on sequences of words or syntactic relations extracted from the parsing tree of sentences. Some of the selected methods focus on more complicated classification algorithms including feature subspace ensembles [34, 25], and a meta-learning model [24].

3 Reproducibility Study

Our reproducibility study consists of seven steps: (1) paper selection, (2) student recruitment, (3) paper assignment and instruction, (4) implementation and experimentation, (5) auditing, (6) publication, and (7) post-publication rebuttal.

(1) *Paper selection.* Every reproducibility study should supply justification for its selection of papers to be reproduced. For example, Ferro and Silvello [13] reproduce a method that has become important for performance measurement in IR in order to raise confidence in its reliability; Hagen et al. [15] reproduce the three best-performing approaches in a shared task, since shared task notebooks are often less well-written than other papers, rendering their reproduction difficult. Other justifications may include: comparison of

a method with one's own approach, doubts whether a particular contribution works as advertised, completing a software library, using an approach as a sub-module to solve a different task, or identifying the best approach for an application.

The goal of our reproducibility study is a certain "coverage" of author identification. Given our limited human resources, we tried to cover different paradigms of author identification, whereas the papers selected were supposed to be influential for the field. In this regard, we considered it vitally important to consult with a domain expert to provide a selection of papers that satisfy these constraints, since hands-on experience is required to make such decisions. Particularly, the various paradigms to solve a problem typically emerge only with hindsight, whereas the terminology used in early papers may differ substantially from the present one. The number of citations that a paper received by itself turns out to be an insufficient yardstick, since this introduces a bias against recent papers. A total of 30 papers have been selected by our domain expert, whereas Table 1 overviews only those that were reproduced by the students recruited.

(2) *Student recruitment.* To scale our reproducibility study, we employ students. Their recruitment for a task like this can be done in various ways within the context of a university, whereas proper incentives should be set for sufficient motivation. A dedicated course or project might be offered, or an extracurricular activity. The latter was what we offered to students from various universities. Altogether, we recruited 16 students with backgrounds in computer science (5), engineering (4), physics (3), and maths (4). Programming experience was in fact the only prerequisite for participation, which is why we did not restrict eligibility to computer science students only.

We were confident that a reproducibility study with students will work, since it resembles everyday work at universities, where advisors often pass tasks to students for implementation under guidance. Moreover, it tells a lot about any given paper whether or not it enables a student with basic training in programming to reproduce its results; ideally, the authors of technical papers ensure that even people outside their domain may follow up on their work. However, most papers omit the basics that are considered folklore in a given discipline, so that we tried to match students by their skill sets to papers, guiding them throughout the process.

(3) *Paper assignment and instruction.* The papers selected by our domain expert are of varying complexity, ranging from basic character-level string processing to dependency parsing to advanced statistical modeling (i.e., a customized LDA approach). Therefore, we did not assign papers at random but based on interviews about backgrounds and programming experiences of our students. More complex papers were assigned to students who have better chances of successfully implementing them. But matching students with papers is non-trivial, since interviews only paint an incomplete picture.

After paper assignment, we handed out papers to students alongside instructions what to do. After a brief explanation of the goals of the study (i.e., reimplementing influential approaches to author identification), the task was specified as follows:

1. Study the proposed main algorithmic contribution for author identification.
2. Implement the approach in a programming language of your choice.
3. Replicate at least one of the experiments described involving the approach.

Further, we asked students to take note of any imprecise, ambiguous, or missing details along the way. We did not ask students to repeat all experiments described in their papers, since we do not suspect the reported results to be false or entirely irreproducible. Rather, we use the papers' experiments as benchmarks to check the students' implementations.

(4) *Implementation and experimentation.* In this step, students worked on their own, but were encouraged to ask questions. Our domain expert was accessible and we discussed technical questions with eleven of the students, most of which pertained to basic text processing, statistical computations, and performance optimization. Since the students lacked background in natural language processing, we pointed them to appropriate libraries that implement things like tokenization and dependency parsing. The students had ample time for implementation and experimentation, however, many started late before the deadline, and one failed to complete his task. To mitigate such issues, we recommend to engage students early on in (teleconference) meetings in this step.

(5) *Auditing.* After implementation, experts and students met for an auditing session. The purpose of this session was to ascertain that students had understood their paper at a fundamental, conceptual level so as to raise confidence in their implementations. Each approach was thoroughly discussed, highlighting the reproducibility issues observed. However, not everyone brought along flawless implementations; due to misunderstandings, some methods had to be amended. Therefore, a hackathon was organized to fix the issues, while encouraging group work and code sharing between compatible implementations. We were accompanying the students at all times during this step. Though we tried to finalize everything during auditing, some things were left for homework.

(6) *Publication.* Open sourcing the code is one of the main points of the exercise in order to provide baseline implementations to both the communities of author identification and information retrieval. We leave the choice of open source license at the discretion of the students. Since publishers are not yet ready to publish material alongside a scientific paper, we publish the code on our own.[2]

(7) *Post-publication rebuttal.* During steps (1)-(6), we specifically avoided to contact the authors of the selected papers. This was to prevent any bias entering our study or being influenced by the authors who might have been anxious about their approaches' performances. After our study has been accepted for publication, the authors were invited for a rebuttal, the outcome of which will be published as material alongside this paper.

4 Reproducibility Report

Each paper was assessed with regard to a number of reproducibility criteria pertaining to (1) approach clarity, (2) experiment clarity and soundness, (3) dataset availability or reconstructability, and (4) overall replicability, reproducibility, simplifiability (e.g., omitting preprocessing steps without harming performance), and improvability (e.g., with respect to runtime). The assessments result from presentations given by the students, a questionnaire, and subsequent individual discussions; Table 2 overviews the results.

(1) *Approach clarity.* For none of the approaches source code (or executables) were available accompanying the papers (only ○ in row "Code available" of Table 2), so that all students had to start from scratch. The students chose the programming language they are

[2] Materials and code of this study are available at www.uni-weimar.de/medien/webis/publications and the latest versions of the code in its GitHub repositories at www.github.com/pan-webis-de (for a convenient overview, see www.github.com/search?q=ECIR+2016+user:pan-webis-de).

Table 2. Assessment of the individual approaches with respect to reproducibility criteria. A ○ indicates lacking reproducibility or information; a ◐ partial reproducibility or information; a ● sufficient reproducibility or information; a – indicates a criterion does not apply. Sizes are indicated as L(arge), M(edium), and S(mall), as judged by our domain expert. Programming languages Python and Java are abbreviated as `Py` and `J`.

Criterion	**Publication**														
	[4]	[5]	[7]	[10]	[12]	[22]	[23]	[24]	[25]	[29]	[32]	[33]	[34]	[35]	[41]
(1) *Approach clarity*															
Code available	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
Description sound	●	●	◐	◐	◐	●	●	●	●	●	●	●	●	●	●
Details sufficient	●	●	◐	◐	◐	●	●	●	●	◐	◐	◐	●	●	●
Paper self-contained	◐	○	●	◐	●	●	◐	●	●	●	○	◐	●	●	●
Preprocessing	○	●	●	●	–	–	–	◐	–	○	○	●	●	–	–
Parameter settings	–	◐	●	◐	●	●	–	●	●	●	●	○	●	●	○
Library versions	–	–	–	○	◐	–	–	◐	–	–	○	○	○	–	–
Reimplementation															
Language	`Py`	`Py`	`Py`	`C++`	`J`	`Py`	`C++`	`Py`	`Py`	`C#`	`C++`	`J`	`Py`	`Py`	`Py`
(2) *Experiment clarity / soundness*															
Setup clear	◐	●	◐	◐	●	●	●	●	●	●	●	◐	●	●	●
Exhaustiveness	◐	○	◐	○	◐	◐	○	◐	●	●	●	◐	●	●	○
Compared to others	○	○	○	●	●	◐	●	●	○	●	●	●	○	◐	●
Result reproduced	◐	◐	○	◐	◐	◐	●	○	◐	●	○	◐	●	●	●
(3) *Dataset reconstructability / availability*															
Text length	L	L	M	S	M	M	M	M	L	M	L	S	M	M	M
Candidate set	M	M	M	S	M	M	L	L	M	M	S	L	M	L	M
Origin given	●	●	◐	○	●	◐	●	●	●	●	●	●	●	●	○
Corpora available	○	○	○	○	●	◐	◐	○	○	●	●	○	●	◐	●
(4) *Overall assessment*															
Replicability	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
Reproducibility	●	◐	●	◐	◐	●	●	●	●	●	○	●	●	●	●
Simplifiability	●	●	●	○	○	○	○	○	○	○	○	○	○	○	●
Improvability	●	●	●	○	○	○	●	○	○	○	○	●	○	○	○

most familiar with, resulting in nine Python reimplementations, four reimplementations in a C dialect, and two Java reimplementations. Keeping in mind that most of the students had not worked in text processing before, it is a good sign that overall they had no significant problems with the approach descriptions. Some questions were answered by the domain expert, while some students also just looked up basic concepts like tokenizing or cosine similarity on their own. The students with backgrounds in math and theory mentioned a lack of formal rigor in the explanations of some papers (indicated by a ◐ in row "Description sound"); however, this was mostly a matter of taste and did not affect the understandability of the approaches. More problematic were two papers for which not even the references contained sufficient information, so that additional sources had to be retrieved by the students to enable them to reimplement the approach. The lack of details on how input should be preprocessed (○ in row "Preprocessing"), what parameter settings were used (○ in row "Parameter settings"), and missing version numbers of libraries employed (○ in row "Library versions") render the replication of seven out of the 15 selected papers' approaches difficult. This had an effect on the perceived approach clarity at an early stage of reimplementation.

(2) *Experiment clarity / soundness.* Since the students were asked to replicate or at least reproduce one of the experiments of their assigned papers, this gave us first-hand insights into the clarity of presentation of the experiments as well as their soundness. The most common problems we found were unclear splits between training and test data (◐ in row "Setup clear"). Another problem was that rather many approaches are evaluated only against simple baselines or only in small-scale experiments (○ in rows "Exhaustiveness" and "Comparison to others"). To rectify this issue, we conduct our own evaluation of all implemented approaches on three standard datasets in Section 5. Altogether, given the influential nature of the 15 selected approaches, it was not unexpected that in twelve cases the students succeeded in reproducing at least one result similar to those reported in the original papers (● or ◐ in row "Result reproduced").

(3) *Dataset availability / reconstructability.* We also asked students and our domain expert to assess the sizes of the originally used datasets. The approaches have been evaluated using different text lengths (S, M, and L indicate message, article, and book size in row "Text length") and different candidate set sizes (S, M, and L indicate below five, below 15, or more authors in row "Candidate set"). In eleven cases, the origin of the data was given, whereas in two cases each, the origin could only be indirectly inferred, or remained obscure. Corpora of which the datasets used for evaluation have been derived were available in four cases, whereas we tried to reconstruct the datasets in cases where sufficient information was given.

(4) *Overall assessment and discussion.* To complete the picture of our assessment, we have judged the overall replicability, reproducibility, simplifiability, and improvability of the original papers. Taking into account papers with only partially available information on preprocessing, parameter settings, and libraries (ten papers) as well as the non-availability of the originally used corpora, none of the 15 publications' results are replicable. This renders the question of at least reproducing the results with a similar approach or using a similar dataset even more important. To this end, students were instructed to use the latest versions of the respective libraries with default parameter settings, and if nothing else helped, apply common sense. Regarding missing information on datasets, our domain expert suggested substitutions. With these remedies, all but one approach achieved results comparable to those originally reported (● or ◐ in row "Reproducibility"). The three partially reproducible papers are due to non-availability of the original data and the use of incomparable substitutions.

Only the reimplementation of the approach of Seroussi et al. [32] has been unsuccessful to date: it appears to suffer from an imbalanced text length distribution across candidate authors, resulting in all texts being attributed to authors with the fewest words among all candidates. This behavior is at odds with the paper, since Seroussi et al. do not mention any problems in this regard, nor that the evaluation corpora have been manually balanced. Since the paper is exceptionally well-written, leaving little to no room for ambiguity, we are unsure what the problem is and suspect a subtle error in our implementation. However, despite our best efforts, we have been unable to find this error to date. Perhaps the post-publication rebuttal phase or future attempts at reproducing Seroussi et al.'s work will shed light on this issue.[3]

In four cases, the respective students, while working on the reimplementations, identified possibilities of simplifying or even improving the original approaches (● in

[3] Confer the repository of the reimplementation of Seroussi et al.'s approach to follow up on this.

Table 3. Evaluation results (classification accuracy) of the reimplemented approaches on three benchmark corpora. Best results (BR) are given as reported by the authors of [1, 12, 21]. Some approaches cannot be applied on all corpora (n/a) for reasons of runtime complexity or insufficient text lengths. One approach could not be successfully reproduced and was hence omitted (–).

Corpus	**Publication**															
	[4]	[5]	[7]	[10]	[12]	[22]	[23]	[24]	[25]	[29]	[32]	[33]	[34]	[35]	[41]	BR
C10	9.0	72.8	59.8	50.2	75.4	71.0	77.2	22.4	72.0	76.6	–	29.8	73.8	70.8	76.6	86.4
PAN11	0.1	29.6	5.4	13.5	43.1	1.8	32.8	n/a	20.2	46.2	–	n/a	7.6	34.5	65.0	65.8
PAN12	85.7	71.4	92.9	28.6	28.6	71.4	n/a	78.6	78.6	57.1	–	n/a	7.1	85.7	64.3	92.9

rows “Simplification” and “Improvability”). A few examples that concern runtime: when constructing the function word graph of Arun et al. [4], it suffices to take only the n last function words in a text window into account, where $n < 5$, instead of all previous ones. In Benedetto et al.’s approach [5], it suffices to only use the compression dictionary of the profile instead of recompressing profile and test text every time. In Burrows’ approach [7], POS-tagging can be omitted, and in the approach of Teahan and Harper [41] one can refrain from actually compressing texts, but just compute entropy. For all of these improvements, the attribution performance was not harmed but often even improved while the runtime was substantially decreased.

On the upside, we can confirm that it is possible to reproduce almost all of the most influential work of a field when employing students to do so. On the downside, however, new ways of ensuring rigorous explanations of approaches and experimental setups should be considered.

5 Evaluation

To evaluate the reimplementations under comparable conditions we use the following corpora:

- *C10.* English news from the CCAT topic of the Reuters Corpus Volume 1 for 10 candidate authors (100 texts each). Best results reported by Escalante et al. [12].
- *PAN11.* English emails from the Enron corpus for 72 candidate authors with imbalanced distribution of texts. The corpus was used in the PAN 2011 shared task [1].
- *PAN12.* English novels for 14 candidate authors with three texts each. The corpus was used in the PAN 2012 shared task [21].

Parameters were set as specified in the original papers, unless they were not supplied, in which case parameters were optimized based on the training data. One exception is the approach of Escalante et al. [12] where a linear kernel was used instead of the diffusion kernel mentioned in that paper, since the latter could not be reimplemented in time.

Table 3 shows the evaluation results. As can be seen, some approaches are very effective on long texts (PAN12) but fail on short (C10) or very short texts (PAN11) [7, 4]. Moreover, some approaches are considerably affected by imbalanced datasets (PAN11) [22]. It is interesting that in two out of the three corpora used (PAN12 and PAN11) at least one of the approaches competes with the best reported results to date. In general, the compression-based models seem to be more stable across corpora probably because they have few or none parameters to be fine-tuned [5, 23, 29, 41]. The best macro-average accuracies on these corpora are obtained by Teahan and Harper [41] and Stamatatos [35]. Both follow the profile-based paradigm which seems to be more robust in case of limited text-length or limited number of texts per author. Moreover, they use character features which seem to be the most effective ones for this task.

6 Conclusion

To the best of our knowledge, a reproducibility study like ours, with the explicit goal of sharing working implementations of many important approaches, is unprecedented in information retrieval and in author identification, if not computer science as a whole. In this regard, we argue that employing students to systematically reimplement influential research and publish the resulting source code may prove to be a way of scaling the reproducibility efforts in many branches of computer science to a point at which a significant portion of research is covered. Conceivably, this would accelerate progress in the corresponding fields, since the entire community would have access to the state of the art. For students in their late education and early careers, reimplementing a given piece of influential research, and verifying its correctness by reproducing experimental results is definitely a worthwhile learning experience. Moreover, reproducing research from fields related to one's own may foster collaboration between both fields involved.

Acknowledgements

This study was supported by the German National Academic Foundation (German: Studienstiftung des deutschen Volkes). The foundation helped to recruit students among its scholars and organized our auditing workshop as part of its 2015 summer academy in La Colle-sur-Loup, France. We thank the foundation for their generous support. Our special thanks go to Dorothea Trebesius, Matthias Frenz, and Martina Rothmann-Stang who provided for our every need at the workshop.

Bibliography

[1] S. Argamon and P. Juola. Overview of the international authorship identification competition at PAN-2011. In *CLEF 2011 Notebooks*.

[2] J. Arguello, F. Diaz, J. Lin, and A. Trotman. *RIGOR @ SIGIR 2015*.

[3] T. G. Armstrong, A. Moffat, W. Webber, and J. Zobel. Improvements that don't add up: Ad-hoc retrieval results since 1998. In *CIKM 2009*, pp. 601-610.

[4] R. Arun, V. Suresh, and C. E. Veni Madhavan. Stopword graphs and authorship attribution in text corpora. In *ICSC 2009*, pp. 192-196

[5] D. Benedetto, E. Caglioti, and V. Loreto. Language trees and zipping. *Phys. Rev. Lett.*, 88: 048702, 2002

[6] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent Dirichlet allocation. *J. Mach. Learn. Res.*, 3: 993-1022, 2003.

[7] J. Burrows. Delta: A measure of stylistic difference and a guide to likely authorship. *Lit. and Ling. Comp.*, 17(3):267-287, 2002.

[8] Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A Library for Support Vector Machines. *ACM TIST*, 2:27:1-27:27, 2011.

[9] C. Collberg, T. Proebstring, and A. M. Warren. Repeatability and benefaction in computer systems research: A study and a modest proposal. TR 14-04, University of Arizona, 2015.

[10] O. de Vel, A. Anderson, M. Corney, and G. Mohay. Mining e-mail content for author identification forensics. *SIGMOD Rec.*, 30(4):55-64, 2001.

[11] E. Di Buccio, G. M. Di Nunzio, N. Ferro, D. Harman, M. Maistro, and G. Silvello. Unfolding off-the-shelf IR systems for reproducibility. In *RIGOR @ SIGIR 2015*.

[12] H. J. Escalante, T. Solorio, and M. Montes-y Gómez. Local histograms of character n-grams for authorship attribution. In *HLT 2011*, pp. 288-298.

[13] N. Ferro and G. Silvello. Rank-biased precision reloaded: Reproducibility and generalization. In *ECIR 2015*, pp. 768-780.

[14] M. Gamon. Linguistic correlates of style: Authorship classification with deep linguistic analysis features. In *COLING 2004*.

[15] M. Hagen, M. Potthast, M. Büchner, and B. Stein. Twitter sentiment detection via ensemble classification using averaged confidence scores. In *ECIR 2015*, pp. 513-525.

[16] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I.H. Witten. The WEKA Data Mining Software: An Update. *SIGKDD Explorations*, 11(1):10-18, July 2009.

[17] A. Hanbury, G. Kazai, A. Rauber, and N. Fuhr. *Proceedings of ECIR 2015*.

[18] D. I. Holmes. The evolution of stylometry in humanities scholarship. *Lit. and Ling. Comp.*, 13 (3):111-117, 1998.

[19] F. Hopfgartner, A. Hanbury, H. Müller, N. Kando, S. Mercer, J. Kalpathy-Cramer, M. Potthast, T. Gollub, A. Krithara, J. Lin, K. Balog, and I. Eggel. Report on the Evaluation-as-a-Service (EaaS) expert workshop. *SIGIR Forum*, 49(1):57-65, 2015.

[20] P. Juola. Authorship attribution. *FnTIR*, 1:234-334, 2008.

[21] P. Juola. An overview of the traditional authorship attribution subtask. In *CLEF 2012 Notebooks*.

[22] V. Keselj, F. Peng, N. Cercone, and C. Thomas. N-gram-based author profiles for authorship attribution. In *PACLING 2003*, pp. 255-264.

[23] D. V. Khmelev and W. J. Teahan. A repetition based measure for verification of text collections and for text categorization. In *SIGIR 2003*, pp. 104-110.

[24] M. Koppel, J. Schler, and E. Bonchek-Dokow. Measuring differentiability: Unmasking pseudonymous authors. *J. Mach. Learn. Res.*, 8:1261-1276, 2007.

[25] M. Koppel, J. Schler, and S. Argamon. Authorship attribution in the wild. *LRE*, 45(1):83-94, 2011.

[26] J. Lin. The open-source information retrieval reproducibility challenge. In *RIGOR @ SIGIR 2015*.

[27] T.C. Mendenhall. The characteristic curves of composition. *Science*, ns-9 (214S):237-246, 1887.

[28] I. Ounis, G. Amati, V. Plachouras, B. He, C. Macdonald, and C. Lioma. Terrier: A High Performance and Scalable Information Retrieval Platform. In *OCIR @ SIGIR 2006*.

[29] F. Peng, D. Schuurmans, and S. Wang. Augmenting naive Bayes classifiers with statistical language models. *Inf. Retr.*, 7(3-4):317-345, 2004.

[30] F. Rangel, P. Rosso, F. Celli, M. Potthast, B. Stein, and W. Daelemans. Overview of the 3rd author profiling task at PAN 2015. In *CLEF 2015 Notebooks*.

[31] Joseph Rudman. The state of authorship attribution studies: Some problems and solutions. *Computers and the Humanities*, 31(4):351-365, 1997.

[32] Y. Seroussi, F. Bohnert, and I. Zukerman. Authorship attribution with author-aware topic models. In *ACL 2012*, pp. 264-269.

[33] G. Sidorov, F. Velasquez, E. Stamatatos, A. Gelbukh, and L. Chanona-Hernández. Syntactic n-grams as machine learning features for natural language processing. *Expert Syst. Appl.*, 41(3): 853-860, 2014.

[34] E. Stamatatos. Authorship attribution based on feature set subspacing ensembles. *Int. Journal on Artificial Intelligence Tools*, 15(5):823-838, 2006.

[35] E. Stamatatos. Author identification using imbalanced and limited training texts. In *DEXA 2007*, pp. 237-241.

[36] E. Stamatatos. A survey of modern authorship attribution methods. *JASIST*, 60:538-556, 2009.

[37] E. Stamatatos, N. Fakotakis, and G. Kokkinakis. Automatic text categorization in terms of genre and author. *Comput. Linguist.*, 26(4):471-495, 2000.

[38] E. Stamatatos, W. Daelemans, B. Verhoeven, B. Stein, M. Potthast, P. Juola, M. A. Sánchez-Pérez, and A. Barrón-Cedeño. Overview of the author identification task at PAN 2014. In *CLEF 2014 Notebooks*.

[39] V. Stodden. The scientific method in practice: Reproducibility in the computational sciences. MIT Sloan Research Paper No. 4773-10, 2010.

[40] N. Tax, S. Bockting, and D. Hiemstra. A cross-benchmark comparison of 87 learning to rank methods. *IPM*, 51(6):757-772, 2015.

[41] W. J. Teahan and D. J. Harper. Using compression-based language models for text categorization. In *Language Modeling for Information Retrieval*, pp. 141-165, 2003.

[42] H. van Halteren. Linguistic profiling for author recognition and verification. In *ACL 2004*, pp. 199-206.

[43] R. Zheng, J. Li, H. Chen, and Z. Huang. A framework for authorship identification of online messages: Writing-style features and classification techniques. *JASIST*, 57(3):378-393, 2006.