A Prototype for Authorship Attribution Studies

Patrick Juola*
juola@mathcs.duq.edu

John Sofko
sofko936@hotmail.com

Patrick Brennan
brennan998@comcast.net
Duquesne University
Pittsburgh, PA 15282
UNITED STATES OF AMERICA

Abstract

Despite a century of research, statistical and computational methods for authorship attribution are neither reliable, well-regarded, widely-used, or well-understood. This paper presents a survey of the current state-of-the-art as well as a framework for uniform and unified development of a tool to apply the state-of-the-art, despite the wide variety of methods and techniques used. The usefulness of the framework is confirmed by the development of a tool using that framework that can be applied to authorship analysis by researchers without a computing specialization. Using this tool, it may be possible both to expand the pool of available researchers as well as to enhance the quality of the overall solutions (for example, by incorporating improved algorithms as discovered through empirical analysis [Juola, 2004a]).

1 Introduction

The task of computationally inferring the author of a document based on its internal statistics – sometimes called "stylometrics," "authorship attribution," or (for the completists) "non-traditional authorship attribution" is an active and vibrant research area, but at present largely without use. For example, unearthing the author of the anonymously-written *Primary Colors* (Joe Klein) became a substantial issue in 1996. In 2004, "anonymous" published *Imperial Hubris*, a followup to his (her?) earlier work *Through Our Enemies' Eyes*. Who wrote these books?[1] Did the same person actually write these books? Does the(?) author actually have the expertise claimed on the dust cover ("a senior

*Corresponding author

[1] According to news report consensus, as first revealed by Jason Vest in the July 2 edition of the Boston Phoenix, the author is Michael Scheuer, a senior CIA officer. But how seriously should we take this consensus?

U.S. intelligence official with nearly two decades of experience")? And, why haven't our computers already given us the answer?

Determining the author of a particular piece of text has been a methodological issue for centuries. Questions of authorship can be of interest not only to humanities scholars, but in a much more practical sense to politicians, journalists, and lawyers as in the examples above. In recent years, the development of improved statistical techniques [Holmes, 1994] in conjunction with the wider availability of computer-accessible corpora [Nerbonne, 2004] has made the automatic inference of authorship (variously called "authorship attribution" or more generally "stylometry") at least a theoretical possibility, and research in this area has expanded tremendously. From a practical standpoint, acceptance of this technology is dogged by many issues — epistemological, technological, and political — that limit and in some cases prevent its wide acceptance. Part of this lack of use can be attributed to simple unfamiliarity on the part of the relevant communities, combined with a perceived history of inaccuracy (see, for example, the discussion of the cusum technique [Farringdon, 1996] in [Holmes, 1998]). Since 1996, however, the popularity of corpus linguistics as a field of study and vast increase in the amount of data available on the Web have made it practical to use much larger sets of data for inference. During the same period, new and increasingly sophisticated techniques have improved the quality (and accuracy) of judgments the computers make.

This paper summarizes some recent findings and experiments and presents a framework for development and analysis to address these issues. In particular, we discuss two major usability concerns, accuracy and user-friendliness. In broad terms, these concerns can only be addressed by expansion of the number of clients (users) for authorship attribution technology. We then present a theoretical framework for description of authorship attribution to make it easier and more practical for the development and improvement of genuine off-the-shelf attribution solutions.

2 Background

With a history stretching to 1887 [Mendenhall, 1887], and 10,700 hits on Google[2], it is apparent that statistical/quantitative authorship attribution is an active and vibrant research area. With nearly 120 years of research, it is surprising that it has not been accepted by relevant scholars : "Stylometrics is a field whose results await acceptance by the world of literary study by and large."[3] This can be attributed at least partially to a limited view of the range of applicability, to a history of inaccuracy, and to the mathematical complexity (and corresponding difficulty of use) of the techniques deployed.

For example, and taking a broad view of "stylometry" to include the inference of group characteristics of a speaker, the story from Judges 12:5–6 describes

[2]Phrasal search for "authorship attribution," June 2, 2005
[3]Anonymous, personal communication to Patrick Juola, 2004

how tribal identity can be inferred from the pronunciation of a specific word (to be elicited). Specifically,

> The Gileadites captured the fords of the Jordan leading to to Ephraim, and whenever a survivor of Ephraim said, "Let me cross over," the men of Gilead asked him, "Are you an Ephraimite?" If he replied, "No," they said, "All right, say 'Shibboleth.'" He said, "Sibboleth," because he could not pronounce the word correctly, they seized him and killed him at the fords of the Jordan. Forty-two thousand Ephraimites were killed at that time.

A more modern version of such *shibboleths* could involve specific lexical or phonological items; a person who writes of a "Chesterfield" as a piece of furniture is presumptively Canadian, and an older Canadian at that [Easson, 2002]. [Wellman, 1936][p. 114] describes how an individual spelling error — an idiosyncratic spelling of "toutch" was elicited and used in court to validate a document for evidence.

At the same time, such tests cannot be relied upon. Idiosyncratic spelling or not, the word "touch" is rather rare (86 tokens in the million-word Brown corpus [Kučera and Francis, 1967]), and although one may be able to elicit it in a writing produced on demand, it's less likely that one will be able to find it independently in two different samples. People are also not consistent in their language, and may (mis)spell words differently at different times; often the tests must be able to handle distributions instead of mere presence/absence judgments. Most worryingly, the tests themselves may be inaccurate [see especially the discussion of CUSUM [Farringdon, 1996] in [Holmes, 1998]], rendering any technical judgment questionable, especially if the test involves subtle statistical properties such as "vocabulary size" or "distribution of function words," concepts that may not be immediately transparent to the lay mind.

Questions of accuracy are of particular importance in wider applications such as law. The relevance of a document (say, an anonymously libelous letter) to a court may depend not only upon who wrote it, but upon whether or not that authorship can be demonstrated. Absent eyewitnesses or confessions, only experts, defined by specialized knowledge, training, experience, or education, can offer "opinions" about the quality and interpretation of evidence. U.S. law, in particular, greatly restricts the admissibility of scientific evidence via a series of epistemological tests[4]. The *Frye* test states that scientific evidence is admissible only if "generally accepted" by the relevant scholarly community, explicitly defining science as a consensus endeavor. Under *Frye*, (widespread) ignorance of or unfamiliarity with the techniques of authorship attribution would be sufficient by itself to prevent use in court. The *Daubert* test is slightly more epistemologically sophisticated, and establishes several more objective tests, including but not limited to empirical validation of the science and techniques used, the existence of an established body of practices, known standards of accuracy (including so-called type I and type II error rates), a pattern of use in

[4] *Frye vs. United States*, 1923; *Daubert vs. Merrill Dow*, 1993.

non-judicial contexts, and a history of peer review and publication describing the underlying science.

At present, authorship attribution cannot meet these criteria. Aside from the question of general acceptance (the quote presented in the first paragraph of this section, by itself, shows that stylometrics couldn't pass the *Frye* test), the lack of standard practices and known error rates eliminates stylometry from *Daubert* consideration as well.

3 Recent developments

To meet these challenges, we present some new methodological and practical developments in the field of authorship attribution. In June 2004, ALLC/ACH hosted an "Ad-hoc Authorship Attribution Competition"[Juola, 2004a] as a partial response to these concerns. Specifically, by providing a standardized test corpus for authorship attribution, not only could the mere ability of statistical methods to determine authors be demonstrated, but methods could further be distinguished between the merely "successful" and "very successful." (From a forensic standpoint, this would validate the science while simultaneously, establishing the standards of practice and creating information about error rates.) Contest materials included thirteen problems, in a variety of lengths, styles, genres, and languages, mostly gathered from the Web but including some materials specifically gathered to this purpose. Two dozen research groups participated by downloading the (anonymized) materials and returning their attributions to be graded and evaluated against the known correct answers.

The specific problems presented included the following:

- *Problem A* (English) Fixed-topic essays written by thirteen Duquesne students during fall 2003.
- *Problem B* (English) Free-topic essays written by thirteen Duquesne students during fall 2003.
- *Problem C* (English) Novels by 19th century American authors (Cooper, Crane, Hawthorne, Irving, Twain, and 'none-of-the-above'), truncated to 100,000 characters.
- *Problem D* (English) First act of plays by Elizabethan/Jacobean playwrights (Johnson, Marlowe, Shakespeare, and 'none-of-the-above').
- *Problem E* (English) Plays in their entirety by Elizabethan/Jacobean playwrights (Johnson, Marlowe, Shakespeare, and 'none-of-the-above').
- *Problem F* ([Middle] English) Letters, specifically extracts from the Paston letters (by Margaret Paston, John Paston II, and John Paston III, and 'none-of-the-above' [Agnes Paston]).
- *Problem G* (English) Novels, by Edgar Rice Burrows, divided into "early" (pre-1914) novels, and "late" (post-1920).

- *Problem H* (English) Transcripts of unrestricted speech gathered during committee meetings, taken from the *Corpus of Spoken Professional American-English*.
- *Problem I* (French) Novels by Hugo and Dumas (pere).
- *Problem J* (French) Training set identical to previous problem. Testing set is one *play* by each, thus testing ability to deal with cross-genre data.
- *Problem K* (Serbian-Slavonic) Short excerpts from *The Lives of Kings and Archbishops*, attributed to Archbishop Danilo and two unnamed authors (A and B). Data was originally received from Alexsandar Kostic.
- *Problem L* (Latin) Elegaic poems from classical Latin authors (Catullus, Ovid, Propertius, and Tibullus).
- *Problem M* (Dutch)
  Fixed-topic essays written by Dutch college students, received from Hans van Halteren.

The contest (and results) were surprising at many levels; some researchers initially refused to participate given the admittedly difficult tasks included among the corpora. For example, Problem F consisted of a set of letters extracted from the Paston letters. Aside from the very real issue of applying methods designed/tested for the most part for modern English on documents in Middle English, the size of these documents (very few letters, today or in centuries past, exceed 1000 words) makes statistical inference difficult. Similarly, problem A was a realistic exercise in the analysis of student essays (gathered in a freshman writing class during the fall of 2003) – as is typical, no essay exceeded 1200 words. From a standpoint of literary analysis, this may be regarded as an unreasonably short sample, but from a standpoint both of a realistic test of *forensic* attribution, as well as a legitimately difficult problem for testing the sensitivity of techniques, these are legitimate.

Results from this competition were heartening. ("Unbelievable," in the words of one contest participant.) Despite the data set limitations, the highest scoring participant [Koppel and Schler, 2004], scored an average success rate of approximately 71%. (Juola's solutions, in the interests of fairness, averaged 65% correct.) In particular, Schler's methods achieved 53.85% accuracy on problem A and 100.00% accuracy on problem F, both acknowledged to be difficult and considered by many to be unsolvably so.

More generally, all participants scored significantly above chance. Perhaps as should be expected, performance on English problems tended to be higher than on other languages. Perhaps more surprisingly, the availability of large documents was not as important to accuracy as the availability of a large number of smaller documents, perhaps because they can give a more representative sample of the range of an author's writing. In particular, the correlation between the average performance of a method on English samples (problems A-H) correlation significantly (0.594, $p < 0.05$) with that method's performance on non-English

samples. Correlation between large-sample problems (problems with over 50,000 words per sample) and small sample problems was still good, although no longer significant (r = 0.3141). This suggests that the problem of authorship attribution is at least somewhat a language- and data-independent problem, and one to which we may be able to expect to find wide-ranging technical solutions for the general case, instead of (as, for example, in machine translation) to have to tailor our solutions with detailed knowledge of the problem/texts/languages at hand. In particular, we offer the following challenge to all researchers in the process of developing a new authorship attribution algorithm : *if you can't get 90% correct on the Paston letters (problem F), then your algorithm is not competitively accurate.* Every well-performing algorithm studied had no difficulty achieving this standard. Statements from researchers that their methods will not work with only a handful of letters as training data should be regarded with appropriate suspicion.

Finally, methods based on simple lexical statistics tended to perform substantially worse than methods based on N-grams or similar measures of syntax in conjunction with lexical statistics. We continue to examine the detailed results in an effort to identify other characteristics of good solutions. Unfortunately, another apparent result is that the high-performing algorithms appear to be mathematically and statistically (although not necessarily linguistically) sophisticated. The good methods have names that may appear fearsome to the uninitiated : linear discriminant analysis [Baayen et al., 2002, van Halteren et al., 2005], orthographic cross-entropy [Juola and Baayen, 2003, Juola and Baayen, 2005], common byte N-grams [Keselj and Cercone, 2004], SVM with a linear kernel function [Koppel and Schler, 2004]. These techniques can be difficult to implement, or even to understand or to use, by a casual, non-technical scholar. At the same time, the sheer number of techniques proposed (and therefore, the number of possibilities available to confuse) has exploded, which also limits the pool of available users. We can no longer expect a casual professor of literature — let alone a journalist, lawyer, judge, or interested layman — to apply these new methods to a problem of interest without technical assistance.

4 New technologies

The variation in these techniques can make authorship attribution appear to be an unorganized mess, but it has been claimed that under an appropriate theoretical framework [Juola, 2004b], many of these techniques can be unified, combined, and deployed. Using this framework, it is possible — indeed, we hope to demonstrate as the basis for incremental improvement — to develop "commercial off the shelf" (COTS) software to perform much of the technical analytic aspects.

The initial observation is that, broadly speaking, all known human languages can be described as an unbounded sequence chosen from a finite space of possible events. For example, the IPA phonetic alphabet [Ladefoged, 1993] describes an

inventory of approximately 100 different phonemes; a typewriter shows approximately 100 different Latin-1 letters; a large dictionary will present an English vocabulary of 50–100,000 different words. An (English) utterance is "simply" a sequence of phonemes (or words).

The proposed framework postulates a three-phase division of the authorship attribution task, each of which can be independently performed, rather in the manner of a Unix or Linux pipeline, where the output of one phase is immediately made available as the input of the following one. These phases are:

- Canonicization — No two physical realizations of events will ever be exactly identical. We choose to treat similar realizations as identical to restrict the event space to a finite set.
- Determination of the event set — The input stream is partitioned into individual non-overlapping "events." At the same time, uninformative events can be eliminated from the event stream.
- Statistical inference — The remaining events can be subjected to a variety of inferential statistics, ranging from simple analysis of event distributions through complex pattern-based analysis. The results of this inference determine the results (and confidence) in the final report.

As an example of how this procedure works, we consider a method for identifying the language in which a document is written. The statistical distribution of letters in English text is well-known (see any decent cryptography handbook, including [Stinson, 2002]). We first canonicize the document by identifying each letter (an italic *e*, a boldface **e**, or a capital E should be treated identically) and producing a transcription. This canonicization process would also implicitly involve other transformations, for example, partitioning a PDF image into text regions to be analyzed as opposed to illustrations and margins to be ignored. A much more sophisticated canonicization process, following [Rudman, 2003], could regularize spelling, eliminate extraneous material such as chapter headings and page numbers, and even "de-edit" the invisible hand of the editor or redactor, to approximate as closely as possible the state of the original manuscript as it left the pen or typewriter of the author. The output of this canonicization process would then be a sequence of linguistic elements.

We then identify each letter as a separate event, eliminating all non-letter characters such as numbers or punctuation. A more sophisticated application might demand instead that letters be grouped into morphemes, syllables, words, and so forth.

Finally, by compiling an event histogram and comparing it with the known distribution, we can determine a probability that the document was written in English. A similar process would treat each *word* as a separate event (eliminating words not found in a standard lexicon) and comparing event histograms with a standardized set such as the Brown histogram [Kučera and Francis, 1967]. Note that the difference between an analysis based on letter histograms and one based on word histograms is purely in the second, event set determination,

phase; the statistics of histogram generation and analysis are identical and can be performed by the same code. The question of the comparative accuracy of these methods can be judged empirically.

The Burrows methods [Burrows, 1989, Burrows, 2003] for authorship attribution can be described in similar terms. After the document is canonicized, the document is partitioned into words-events. Of the words, most words (except for a chosen few function words) are eliminated. The remaining word-events are collected in a histogram, and compared statistically via principle content analysis (PCA) to similar histograms collected from anchor documents. (The difference between the 1989 and 2003 methods is simply in the nature of the statistics performed.)

Even Wellman's "toutch" method can be so described; after canonicization, the event set of words is compiled, specifically, the number of words spelled "toutch." If this set is non-empty, the document's author is determined.

This framework also allows researchers both to focus on the important differences between methods and to mix and match techniques to achieve the best practical results. For example, [Juola and Baayen, 2005] describes two techniques based on cross-entropy that differ only in their event models (words vs. letters). Presumably, the technique would also generalize to other event models (function words, morphemes, parts of speech), and and similarly other inference techniques would work on a variety of event models. It is to be hoped that from this separation, researchers can identify the best inference techniques and the best models in order to assemble a sufficiently powerful and accurate system.

5 Demonstration

The usefulness of this framework can be shown in a newly-developed user-level authorship attribution tool. This tool coordinates and combines (at this writing) several different technical approaches to authorship attribution [Burrows, 1989, Juola, 1997, Burrows, 1989, Kukushkina et al., 2000, Juola, 2003b, Keselj and Cercone, 2004].

Written in Java, this program combines a simple GUI atop the three-phase approach defined above. Users are able to select a set of sample documents (with labels for known authors) and a set of testing documents by unknown authors. The three-phase framework described above fits well into the now standard modular software design paradigm using Java's object-oriented framework. Each of the individual phases is handled by a separate class/module that can be easily extended to reflect new research developments

The original JGAAP[5] prototype was developed in July, 2004. It served as a proof of concept for automating authorship attribution technologies. Unfortunately, the prototype was not developed with extensibility in mind. The architecture used was not clearly defined and the application was not easily modified. These design issues were addressed in the second (current) version of JGAAP. Nearly all of the original source code was refactored to conform to the

[5] *Java Graphical Authorship Attribution Program*; the authors invite suggestions for a better name for future versions.

new design framework. The new JGAAP framework is devised from a strongly object oriented perspective. The core functionality of JGAAP is distilled into seven basic operations. These seven operations include:

- Core Classes
- Document Input
- Creating Events
- Document Preprocessing
- Document Scoring
- Displaying Results
- and Graphical User Interface

The directory structure of the application reflects these operations, making the source code easy to follow and understand.

Core Classes As the name implies, the Core Classes provide the basic framework of the application. By themselves, they provide no application functionality. They are necessary, however, when implementing Java Interfaces to extend functionality.

Document Input The document input module provides methods for importing documents into JGAAP. Currently, JGAAP provides input from local files only, although ongoing improvements to accept documents by remote file transfer or from the Web are in the process of being added.

Creating Events The events module modifies the input documents prior to scoring. These events specify the means by which the documents are presented to the scoring method. Currently, JGAAP provides two types of events: Letters or Words.

Document Preprocessing The document preprocessing module provides methods of modifying the documents prior to scoring as detailed above. Currently, we have made available the following preprocessing options: Removing End Punctuation, Removing HTML Tags, Removing Non-Letters, Removing Numerals (and replacing them with a <NUM> tag), Removing Spaces, and Conversion of Documents to Lower Case.

Document Scoring The document scoring module contains methods for document comparison. These methods apply authorship attribution techniques to compare the input documents and provide a quantitative score for each comparison.

Displaying Results This module contains implementations that are utilized to display scoring results to the end user. The scoring methods currently output a matrix that contains the result of comparing each unknown document with all documents of known authorship. Code within this module may reformat this information into a visual representation of the matrix. Currently, JGAAP provides output of the matrix to the console, to file, or via message box.

Graphical User Interface This module contains the methods responsible for creating the user interface of JGAAP.

The user is able to select from a menu of event selection/preprocessing options and of technical inference mechanisms. Specifically, we designed a multi-menu, panel-based GUI that resembles Microsoft software to facilitate ease of use. The menus are clearly marked and set up so the flow of work is fairly linear and maps closely to the phase structure described above. The documents to be analyzed and the pre-processing and methods of the analysis are selected by the user, then that data gets sent to the (computational) "backend", which returns the results back to the GUI to be displayed.

There are still a number of substantial issues to address in further versions of JGAAP, including improvement of existing factors and the development of new features.

First, we are unsatisfied with the saving/loading method currently implemented into JGAAP. While it is functional, it relies on absolute path names, so it is not especially flexible as we should like, and specifically is restricted only to local files. We would like to add the capability of dynamic path-based "manifest" files in folders of documents. When you load the manifest, you would only have to point JGAAP to where the folder of files is located and it would do the rest of the work for you. We also hope to incorporate state-based processing, where the program generates a a static list that loads along with the program while it starts a new session. This list would have on it all the documents that have been previously input into the program along with the program saving local copies of the tests. When a user wishes to analyze documents, he can select which documents he wishes to check from the permanent list, instead of having to load in the documents every time he wishes to analyze them. While it might almost mean a complete re-design of the GUI from the ground up, it could drastically improve functionality for users that check the same documents over and over. I would like to get the opinions of the community as a whole, because it might not be all that useful.

We hope to get opinions from the community on how they would like to see the data graphically interpreted and displayed. Because this is being developed for the community as a whole, it is important for them to have feedback on how they would like to see the data presented.

Finally, we wish to add a wizard mode and in-context help files to assist new users to the JGAAP program. As more features get added, the complexity of the program will warrant helping the user as much as possible; especially if the idea is to make the program suitable for the general user. Parties interested in seeing or using this program, and especially in helping with the necessary feedback, should contact the corresponding author.

6 Design Issues

The framework outlined above relies heavily on the Java concept of *Interface*. A Java Interface provides a powerful tool that can be used to create highly exten-

sible application frameworks. Conceptually speaking, an interface is a defined set of functions (formally, "methods")that a piece of code can "implement" using any algorithm desired. This permits other pieces of code to use differing variations of the same interface with no changes, permitting easy updates as new techniques are developed and implemented.

Within the Interfaces directory of the application, there exists five defined interfaces: Display, Event, Input, Preprocess, and Score. These interfaces specify required methods that must exist in classes that intend to implement the respective interfaces. The classes within the Core Classes directory contain methods that accept interfaces as parameters. For example, we will assume that a future developer wants to create a new method to display score output. According to the Display interface, the new method may implement Display if and only if it contains the *public void display()* method. The core class Display contains a *public void display(DisplayInterface display)* method. This method accepts an object of type DisplayInterface as a parameter and calls that object's *public void display()* method. Conversely, any code with a *public void display()* method can be called as a Display interface, so a technically sophisticated user who wants to see dendrograms as output need only write a single function, one that takes the matrix results from document scoring and computes (and displays) an appropriate dendrogram. This function can be added on the fly to the JGAAP program and further can be re-used by others, irrespective of the different choices they may have made about the documents, the event model, or the statistics.

Similarly, preprocessing can be handled by separate instantiations and subclasses. Even data input and output can be modularized and separated. As written, the program only reads files from a local disk, but a relatively easy modification (in progress) would allow files to be read from the network (for instance, Web pages from a site such as Project Gutenberg or *literature.org*).

7 Discussion and Future Work

From initial impressions, this tool is both usable and fulfills part of the need of non-technical researchers interested in authorship attribution. On the other hand, this tool is clearly a "research-quality" prototype, and additional work will be needed to implement a wide variety of methods, to determine and implement additional features, to establish a sufficiently user-friendly interface. Even questions such as the preferred method of output — dendrograms? MDS subspace projections? Fixed attribution assignments as in the present system? — are in theory open to discussion and revision. It is hoped that the input of research and user groups such as the present meeting will help guide this development.

Most importantly, the availability of this tool (which we hope will spur additional research by the interested but computationally unsophisticated) should also spur discussion of the role to be played by commercial, off the shelf (COTS) attribution software. As discussed in depth by [Rudman, 2003], authorship at-

tribution is a very nuanced process when properly done. Ideally, as Rudman's Law puts it, *the closest text to the holograph should be found and used.* The editor's pen, the typist's fingers, and the printer's press can all introduce errors – and when a document exists only in physical or image form, the errors introduced by an OCR process [Juola, 2003b] can entirely invalidate the results. Only if all of the analytic and control texts are valid can the results be trusted. This includes not only issues of authenticity, but also of representativeness – if an author's style changes over time [Juola, 2003a, Juola, 2006] a work from outside the period of study will be unrepresentative and may poison the analytic well. Similarly, texts with extensive quotation may be more represented of the quoted sources than of the official author. Texts from the Internet in particular may well be regarded with suspicion due to the poor quality control of Internet publishing in general.

Only once a suitable test suite has been developed can the computational analysis truly proceed, but even here, there are possible pitfalls. The analyst should also be aware of some of the issues introduced by the computational tool. For example, JGAAP uses a fairly simple (and naive) definition of a "word" — a maximal non-blank string of characters. This means that some items may be treated as multiple words ("New" "York" "City") while others are treated as a single word ("non-blank"). An analysis based on part-of-speech types [Juola and Baayen, 2005] will depend upon the accuracy of POS tagger as well as on its tag set. Such subtle distinctions will almost certainly have an effect in some analyses and be entirely irrelevant in others. The computer, of course, is blissfully ignorant of such nuances and will happily analyze the most appalling garbage imaginable. A researcher who accepts such garbage as accurate — Garbage In, Gospel Out — may be said to deserve the consequences. But the client of a lawyer wrongly convicted on such weak evidence deserves better.

Have we, then, made a Faustian bargain in creating such a "plug and play" authorship attribution system? We hope not. The benefits from the wide availability of a tool to the reasoned and cautious researchers who will benefit from it should outweigh the harm caused by misuse in the hands of the injudicious. It is, however, appropriate to consider what sort of safeguards might be created and to what extent the program itself may be able to incorporate and to enforce automatically these safeguards.

From a broader perspective, this program provides a uniform framework under which competing theories of authorship attribution can both be compared and combined (to their hopefully mutual benefit). It also form the basis of a simple user-friendly tool to allow users without special training to apply technologies for authorship attribution and to take advantage of new developments and methods as they become available. From a standpoint of practical epistemology, the existence of this tool should provide a starting point for improving the quality of authorship attribution as a forensic examination – by allowing the widespread use of the technology, and at the same time providing an easy method for testing and evaluating different approaches to determine the necessary empirical validation and limitations.

References

[Baayen et al., 2002] Baayen, R. H., van Halteren, H., Neijt, A., and Tweedie, F. (2002). An experiment in authorship attribution. In *Proceedings of JADT 2002*, pages 29–37, St. Malo. Université de Rennes.

[Burrows, 2003] Burrows, J. (2003). Questions of authorships : Attribution and beyond. *Computers and the Humanities*, 37(1):5–32.

[Burrows, 1989] Burrows, J. F. (1989). 'an ocean where each kind. . . ' : Statistical analysis and some major determinants of literary style. *Computers and the Humanities*, 23(4-5):309–21.

[Easson, 2002] Easson, G. (2002). The linguistic implications of shibboleths. In *Annual Meeting of the Canadian Linguistics Association*, Toronto, Canada.

[Farringdon, 1996] Farringdon, J. M. (1996). *Analyzing for Authorship : A Guide to the Cusum Technique*. University of Wales Press, Cardiff.

[Holmes, 1994] Holmes, D. I. (1994). Authorship attribution. *Computers and the Humanities*, 28(2):87–106.

[Holmes, 1998] Holmes, D. I. (1998). The evolution of stylometry in humanities computing. *Literary and Linguistic Computing*, 13(3):111–7.

[Juola, 1997] Juola, P. (1997). What can we do with small corpora? Document categorization via cross-entropy. In *Proceedings of an Interdisciplinary Workshop on Similarity and Categorization*, Edinburgh, UK. Department of Artificial Intelligence, University of Edinburgh.

[Juola, 2003a] Juola, P. (2003a). Becoming Jack London. In *Proceedings of QUALICO-2003*, Athens, GA.

[Juola, 2003b] Juola, P. (2003b). The time course of language change. *Computers and the Humanities*, 37(1):77–96.

[Juola, 2004a] Juola, P. (2004a). Ad-hoc authorship attribution competition. In *Proc. 2004 Joint International Conference of the Association for Literary and Linguistic Computing and the Association for Computers and the Humanities (ALLC/ACH 2004)*, Göteborg, Sweden.

[Juola, 2004b] Juola, P. (2004b). On composership attribution. In *Proc. 2004 Joint International Conference of the Association for Literary and Linguistic Computing and the Association for Computers and the Humanities (ALLC/ACH 2004)*, Göteborg, Sweden.

[Juola, 2006] Juola, P. (2006). Becoming Jack London. *Journal of Quantitative Linguistics*.

[Juola and Baayen, 2003] Juola, P. and Baayen, H. (2003). A controlled-corpus experiment in authorship attribution by cross-entropy. In *Proceedings of* ACH/ALLC-2003, Athens, GA.

[Juola and Baayen, 2005] Juola, P. and Baayen, H. (2005). A controlled-corpus experiment in authorship attribution by cross-entropy. *Literary and Linguistic Computing*, 20:59–67.

[Keselj and Cercone, 2004] Keselj, V. and Cercone, N. (2004). CNG method with weighted voting. In Juola, P., editor, *Ad-hoc Authorship Attribution Contest*. ACH/ALLC 2004.

[Koppel and Schler, 2004] Koppel, M. and Schler, J. (2004). Ad-hoc authorship attribution competition approach outline. In Juola, P., editor, *Ad-hoc Authorship Attribution Contest*. ACH/ALLC 2004.

[Kukushkina et al., 2000] Kukushkina, O. V., Polikarpov, A. A., and Khmelev, D. V. (2000). Using literal and grammatical statistics for authorship attribution. *Problemy Peredachi Informatii*, 37(2):96–198. Translated in "Problems of Information Transmission," pp. 172–184.

[Kučera and Francis, 1967] Kučera, H. and Francis, W. N. (1967). *Computational Analysis of Present-day American English*. Brown University Press, Providence.

[Ladefoged, 1993] Ladefoged, P. (1993). *A Course in Phonetics*. Harcourt Brace Jovanovitch, Inc., Fort Worth, 3rd edition.

[Mendenhall, 1887] Mendenhall, T. C. (1887). The characteristic curves of composition. *Science*, IX:237–49.

[Nerbonne, 2004] Nerbonne, J. (2004). The data deluge. In *Proc. 2004 Joint International Conference of the Association for Literary and Linguistic Computing and the Association for Computers and the Humanities (ALLC/ACH 2004)*, Göteborg, Sweden. To appear in *Literary and Linguistic Computing*.

[Rudman, 2003] Rudman, J. (2003). On determining a valid text for non-traditional authorship attribution studies : Editing, unediting, and de-editing. In *Proc. 2003 Joint International Conference of the Association for Computers and the Humanities and the Association for Literary and Linguistic Computing (ACH/ALLC 2003)*, Athens, GA.

[Stinson, 2002] Stinson, D. R. (2002). *Cryptography: Theory and Practice*. Chapman & Hall/CRC, Boca Raton, 2nd edition.

[van Halteren et al., 2005] van Halteren, H., Baayen, R. H., Tweedie, F., Haverkort, M., and Neijt, A. (2005). New machine learning methods demonstrate the existence of a human stylome. *Journal of Quantitative Linguistics*, 12(1):65–77.

[Wellman, 1936] Wellman, F. L. (1936). *The Art of Cross-Examination*. MacMillan, New York, 4th edition.