A Prototype for Authorship Attribution Software

Patrick Juola *(juola@mathcs.duq.edu)*
Duquesne University

The task of computationally inferring the author of a document based on its internal statistics — sometimes called *sylometrics*, *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* 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? Does the author actually have the expertise claimed on the dust cover ('a senior U.S. intelligence official with nearly two decades of experience')? And, why haven't our computers already given us the answer?

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 [1]. 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 (Nerbonne) 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 judgements the computers make.

As a recent example, in June 2004, *ALLC/ACH* hosted an *Ad-hoc Authorship Attribution Competition* (Juola 2004b). 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', and analyzed in particular into possible areas of individual success.

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. Despite this extreme paucity of data, results could be stunningly accurate. The highest scoring participant was the research group of Vlado Keselj, with an average success rate of approximately 69%. (Juola's solutions, in the interests of fairness, averaged 65% correct.) In particular, Keselj's methods achieved 85% accuracy on problem A and 90% accuracy on problem F, both acknowledged to be difficult and considered by many to be unsolvably so.

However, the increased accuracy has come at the price of decreased clarity; the statistics used [2] can be hard to understand, and perhaps more importantly, difficult to implement or to use by a non-technical scholar. At the same time, the sheer number of techniques proposed (and therefore, the number of possibilities available to confuse) has exploded. This limits the pool of available users, making it less likely that a casual scholar — let alone a journalist, lawyer, or interested layman — would be able to apply these new methods to a problem of real interest.

I present here a prototype and framework for a user-friendly software system (Juola & Sofko) allowing the casual user to apply authorship attribution technologies to her own purposes. It combines a generalized theoretical model (Juola, 2004b) built on an inference task over *event* sequences with an extensible, object-oriented inference engine that makes the system easily updatable to incorporate new technologies or to mix-and- match combinations of existing ones. The model treats linguistic (or paralinguistic) data as a sequence of separable user-defined *events*, for instance, as a sequence of letters, phonemes, morphemes, or words. These sequences are treated to a three-phase process:

- 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 illustration, the implementation of these phases for the Burrows method would involve, first, *canonicization* by norming the documents of interest. For example, words with variant capitalization (*the*, *The*, *THE*) would be treated as a single type. More sophisticated canonicization procedures could regularize spelling, eliminate extraneous material such as chapter headings, or even "de-edit" (Rudman) the invisible hand of the editor.

During the second phase, the appropriate set of function words would be determined and presented as a sequence of events, eliminating words not in the set of interest. Finally, the appropriate function words are tabulated (without regard to ordering) and the appropriate inferential statistics (principle component analysis) performed. However, replacement of the third stage (and only the third stage) by a linear discriminant analysis would produce a different technique (Baayen et al.).

This framework fits well into the now-standard modular software design paradigm. In particular, the software to be demonstrated uses the Java programming language and object-oriented design to separate the generic functions of the three phases as individual classes, to be implemented as individual subclasses.

The user can select from a variety of options at each phase, and the system as a whole is easily extensible to allow for new developments. For example, the result of event processing is simply a Vector (Java class) of events. Similarly, similarity judgement is a function of the Processor class, which can be instantiated in a variety of different ways. At present, the Processor class is defined with a number of different methods[3]. A planned improvement is to simply define a calculateDistance() function as part of the Processor class. The Processor class, in turn, can be subclassed into various types, each of which calculates distance in a slightly different way.

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 would allow files to be read from a local disk or from the network (for instance, Web pages from a site such as *Project Gutenberg* or literature.org). Users can therefore select functionality as needed on a module-by-module basis both in terms of feature as well as inference method; the current system incorporates four different approaches (Burrows; Juola 1997; Kukushkina et al.; Juola 2003).

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 valididation and limitations.

On the other hand, this tool is also 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 such as the present meeting will help guide this development.

---

1. See, for example, the discussion of the cusum technique (Farrington) in (Holmes 1998).
2. E.g. linear discriminant analysis of common function words (Burrows, Baayen et al; Juola & Baayen), orthographic cross-entropy (Juola, 1996), common byte N-grams (Keselj, 2004).
3. For example, `crossEntDistance()` and `LZWDistance()`.

Bibliography

Baayen, R. H., H. Van Halteren, and F. Tweedie. "Outside the cave of shadows: Using syntactic annotation to enhance authorship attribution." *Literary and Linguistic Computing* 11 (1996): 121-131.

Baayen, R. H., H. Van Halteren, A. Neijt, and F. Tweedie. "An experiment in authorship attribution." *Proceedings of JADT 2002.* St. Malo, 2002. 29-37.

Burrows, J. "'An Ocean where each Kind. . . ': Statistical analysis and some major determinants of literary style." *Computers and the Humanities* 23.4-5 (1989): 309-21.

Burrows, J. "Questions of authorship : Attribution and beyond." *Computers and the Humanities* 37.1 (2003): 5-32.

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

Holmes, D. I. "Authorship attribution." *Computers and the Humanities* 28.2 (1994): 87-106.

Holmes, D. I. "The evolution of stylometry in humanities computing." *Literary and Linguistic Computing* 13.3 (1998): 111-7.

Juola, P. "What can we do with small corpora? Document categorization via cross-entropy." *Proceedings of an Interdisciplinary Workshop on Similarity and Categorization.* Edinburgh, UK: Department of Artificial Intelligence, University of Edinburgh, 1997. n. pag.

Juola, P. "The time course of language change." *Computers and the Humanities* 37.1 (2003): 77-96.

Juola, P. "Ad-hoc authorship attribution competition." *Proceedings of the 2004 Joint International Conference of the Association for Literary and Linguistic Computing and the Association for Computers and the Humanities (ALLC/ACH 2004).* Goteborg, Sweden, 2004a. 175-176.

Juola, P. "On composership attribution." *Proceedings of the 2004 Joint International Conference of the Association for Literary and Linguistic Computing and the Association for Computers and the Humanities (ALLC/ACH 2004).* Goteborg, Sweden, 2004b. 79-80.

Juola, P., and H. Baayen. "A controlled-corpus experiment in authorship attribution by cross-entropy." *Proceedings of ACH/ALLC-2003.* Athens, GA, 2003. n. pag.

Juola, P., and J. Sofko. "Proving and Improving Authorship Attribution Technologies." *Proceedings of CaSTA-2004.* Hamilton, ON, 2004. n. pag.

Keselj, V., and N. Cercone. "CNG Method with Weighted Voting." *Ad-hoc Authorship Attribution Contest Technical Report.*

Kucera, H., and W.N. Francis. *Computational Analysis of Present-day American English.* Providence: Brown University Press, 1967.

Kukushkina, O.V., A.A. Polikarpov, and D.V. Khmelev. "Using literal and grammatical statistics for authorship attribution." *Problemy Peredachi Informatii* 37.2 (2000): 172-184. Translated in *Problems of Information Transmission*.

Nerbonne, J. "The data deluge." *Literary and Linguistic Computing* (Forthcoming). [In Proceedings of the 2004 Joint International Conference of the Association for Literary and Linguistic Computing and he Association for Computers and the Humanities (ALLC/ACH 2004).]