Edinburgh and the Hidden Roots of Darwinism: A Peek Behind a Veil of Anonymity: Supporting Information

K. Tanghe & M. Kestemont

March 2, 2017

The repository associated with this paper contains all the data and software to repeat our experiments.[1] It contains the code as a single-file Python script with minimal dependencies. Minor details (e.g. exact hyperparametrization) can be verified there, and we restrict the discussion in this appendix here to the main aspects of our approach. Some general aspects of the descriptions in this appendix have been copied from the (SI to the) following previously published works:

- Koppel M & Winter Y. Determining if two documents are written by the same author. *J Assoc Inf Sci Technol*, 2014;65(1), 178?187.
- Kestemont M, Stover J, Koppel M, Karsdorp K & Daelemans W. Authenticating the writings of Julius Caesar. *Expert Syst Appl* 2016;63(3), 86-96.

1 Data and preprocessing

Our analyses depart from UTF8 encoded plain text files. These containing machine-readable versions of the original publications, mostly automatically digitized via optical character recognition. The materials have been drawn from a variety of sources (e.g. Google Books etc.). The authorship of the majority of these texts is known and to the best of our knowledge not disputed. The imposter verification algorithm outlined above crucially depends on the availability of distractor documents, which are close enough to the foreground corpus to serve as useful comparands. The imposter texts in our collection are contemporary to the anonymous documents under scrutiny and they are similar to them in topic, genre and themes. The foreground collection includes a selection of texts by scientific writers who have been explicitly named in the literature as suitable candidate authors for the anonymous documents.

The texts have been stripped of front and back matter, including titles, author names and signatures, revealing for instance a text's date of publication. Other than that, no attempt has been made to remove other named entities etc. from the running texts. The materials nevertheless remain noisy, which is mainly due the OCR process. Previous

[1] `https://github.com/mikekestemont/edinburgh`.

research has nevertheless demonstrated that such noise does not pose a major impediment in computational authorship studies: most algorithms prove to be surprisingly robust to, for instance, lingering OCR errors in texts. We have attempted to restore word splitted across linebreaks. All texts were lowercased, and tokenized using the Natural Language ToolKit. Editorial interference cannot be ruled out, but this generally not believed to present any unsurmountable issues for the anonymous texts studied here.

From the writings collected for this paper, only documents were eventually considered which consisted of at least 1,000 tokens (after tokenization). Documents longer than 3,349 tokens (i.e. the length of the main text under scrutiny here: AnonA, 'Observations on the nature and importance of geology') were divided into consecutive, non-overlapping samples of 3,349 tokens. This segmentation procedure served to normalized the skewedness of text length in the materials. The other anonymous text under scrutiny here (AnonB, Of the changes which life has experienced on the globe) only amounted to 1,093 tokens. As detailed in the main text, the authors included in the foreground corpus are: Jameson, Bouó, Grant', Weaver', Fleming and Lyell.

1.1 Imposters Verification

In this section, we discuss our verification algorithm. Traditional verification systems resort to the calculation of a direct, 'first-order' distance measure (e.g. the cityblock metric) between two documents to assess whether they are similar enough to be attributed to the same individual. The imposters method, however, calculates a 'second-order' metric (see Algorithm 1). Let x be the vector representing an anonymous document which is compared to $T = \{t_1, \ldots, t_n\}$, a set of document by the target author. The task is to determine whether the documents in T were or were not written by the author as x. Additionally, the procedure has access to $I = \{i_1, \ldots, i_n\}$, a set of distractor document by so-called imposter authors. The algorithm then starts a bootstrapped procedure: during $k = 1,000$ iterations, it randomly samples a subset of the available features (50%) as well as a random subset of $m = 30$ imposters from I as I'. In each iteration, we determine whether x is closer than any of the documents in T than in I', given the impaired feature space and a distance function $dist$. Instead of returning a first-order distance, the algorithm returns a second-order metric ($score$) indicating the proportion of iterations in which x was closer to an item in T than in I'. As a proportion, the returned second-order $score$ will lie between 0 and 1. No score shifting was necessary as in our previous work.

Algorithm 1: General Imposters Method

```
input : $x$, an anonymous document; $T = \{t_1, \dots, t_n\}$, a set of document by the
        target author; $I = \{i_1, \dots, i_n\}$, a set of documents by other authors;
output: $0 <= score <= 1$

Set $score$ = 0;
for $i \leftarrow 1$ to $k$ do
    Randomly select $rate\%$ of the available features;
    Randomly select $m$ imposters from $I$ as $I'$;
    if $min(dist(T, x)) < min(dist(I', x))$ then
        $score$ = $score + 1/k$ ;
end
Return $1 - score$;
```

1.2 Vector space model and distance metric

The simple *term frequency* model for corpus representation defines term frequency at the document level, as a scalar value representing the ratio of the term's absolute frequency to the total number of terms in a document. Given a fixed vocabulary of size *n*, each document in the collection can be represented as a fixed-length numerical feature vector in the corpus vector space, for some fixed ordering of the features:

$$\langle \mathrm{tf}_1, \mathrm{tf}_2, \dots, \mathrm{tf}_n \rangle \tag{1}$$

This model forms the basis for the inverse-document frequency model (*tf·idf*) which we use, a well-known term weighting scheme from Information Retrieval. This model captures how specific a term is to a specific document and re-score a term's *tf* by weighing it with the term's inverse document frequency (*idf*). For a feature at index i in such a model; the traditional *idf* is taken to be the logarithm of the ratio of the total number of documents in the corpus (*N*) to the number of documents the feature *f* appears in ($\mathrm{df_i}$):

$$idf_i = \log\left(\frac{N}{df_i)}\right) \tag{2}$$

We weigh a_i to obtain its *idf*-weighted counterpart (*tf·idf(a_i)*):

$$\mathrm{tfidf}(a_i) = \mathrm{df}(a_i) \times \mathrm{idf}(a_i) \tag{3}$$

Finally, the resulting vector are scaled to unit norm (L2 normalisation).

Let a and b represent two document vectors, consisting of n features in some fixed order. Let a_i and b_i represent the value of the i-th feature in both documents respectively. We use the minmax measure to calculate the stylistic distance between them:

$$minmax(a, b) = 1 - \left(\frac{\sum\limits_{i=1}^{n} \min(a_i, b_i)}{\sum\limits_{i=1}^{n} \max(a_i, b_i)}\right) \tag{4}$$

We implement the *minmax* measure as a distance metric, instead of a similarity measure. We apply this distance function only to the vocabulary items which are present in the unknown document.

We base our vector space model on the following features: word unigrams and character tetragrams. With features are limited to the 50,000 most frequently occurring items in the entire data set (i.e. the dimensionality of the full document vectors is 100,000). Our previous research offers plenty indications that these language-independent feature types suffice.