Appendix B

Publications
Investigating Hybrids of Evolutionary Search and Linear Discriminant Analysis for Authorship Attribution

Kareem Shaker, David Corne, and Richard Everson

Abstract—Authorship Attribution is the problem of determining who is (or was) the author of one or more texts, in cases where authorship is disputed. There are many well-known cases of disputed authorship; in this paper we consider the Federalist Papers, and the 15th Book of Oz. We treat the problem as a supervised classification problem, and use evolutionary algorithms to search through subsets of function words, which in turn form the basis of predicting authorship via linear discriminant analysis. We compare two approaches (due to the size of the text corpora in dispute, extensive experimentation is difficult), both centred around the optimization of ROC curves. On both datasets, the hybrid EA approach was able to classify the disputed works with 100% accuracy, using small sets of function words comparable to or better than previous works on these cases.

INTRODUCTION

The Authorship Attribution problem is the challenging task of determining who was the author of a disputed text, given that two or more individuals claim sole authorship of that text. There are many historical examples relating to conflicting authorship claims. Two well-known cases, which we use as test cases in this paper, are the disputed Federalist papers [1] and the 15th Book of Oz [2].

As noted in a recent survey [3], a wide variety of approaches have been attempted for authorship attribution, but no specific approach emerges as well-regarded or well-used in this field. Typical approaches attempt to find patterns that characterise specific authors. These patterns may involve, for example, distributions of word lengths, of sentence lengths, and/or vocabulary distribution (in which is considered the overall diversity of terms used by an author).

A particularly popular and successful source of such characteristic patterns is the frequencies of so-called function words. This term was introduced in [1], in which Mosteller and Wallace suggested that an author’s essential style could be characterised by the frequencies with which they used a relatively small number of specific words. In several studies since, function words have been found to be surprisingly useful in authorship attribution studies using a wide range of methods [4]. The primary list of 70 function words studied by Mosteller and Wallace [1] are provided in Table I.
Typically, for a given authorship attribution task, researchers will choose a set of appropriate function words, and construct datasets consisting of frequency vectors for each of several texts known to be have been authored by the disputants. A statistical approach and/or a machine learning method is then used to train a classifier on these data, and this classifier is then applied to frequency vectors representing the disputed text(s). Many classification methods have been tried, ranging through neural networks [5] and support vector machines [6], joining a rich history of statistical and probabilistic approaches [e.g. 7, 8, 9].

A familiar theme in this research area is that of a semi-independent interest in both overall classification accuracy, and in the number of function words (and or other markers) required for an accurate classifier. Focus on the latter issue is of interest and importance in the wider science of stylometry [10], which considers the general question of numerical and statistical patterns that capture an author’s (or a composer’s) style. For example, in one study that compared several approaches on the Federalist Papers task [11], an evolutionary algorithm was employed to evolve rules that queried the frequencies of specific function words, and applied to the Federalist Papers task. Though not always successful in classifying the disputed papers, the results were promising and interesting in that rules were found that used a relatively small number of function words to achieve discrimination.

Given these paired themes of accuracy and feature selection, work on authorship attribution is beginning to emerge that combines the two in the expected variety of ways. For example, most work to date can be seen as involving a priori feature selection, in which researchers have pre-chosen a set of function words, and trained a classifier based on their frequencies. Meanwhile [12] uses an evolutionary algorithm for combined feature selection and classification, using the strategy (increasingly common in general) of a pre-chosen classifier (in their case, a support vector machine), and using an evolutionary algorithm to search the space of feature subsets for input to that classifier. The only other work we have found using evolutionary algorithms in this area is a preliminary report [13].

In this work we contribute a similar approach (that is new to the field of authorship attribution), in which we evolve feature subsets for classification by linear discriminant analysis (LDA), using area-under-ROC-curve as our fitness measure following the training of the classifier. The approach involves careful steps to ensure good generalisation performance, and we find that, for both the cases of the Federalist papers and the Book of Oz, this approach is able to yield discriminators that perfectly classify the disputed works, using numbers of function words than compare favourably with the literature.

The remainder is set out as follows. In section II we provide further background on Authorship Attribution, and in section III we expand on the Federalist and Oz datasets and the way that we process them. Section IV details the linear discriminant analysis (LDA) classifier that we use, with associated information about ROC curves, and section V describes two hybrid evolutionary algorithms that we use to evolve feature subsets for this classifier. Experiments and results are given in section VI, and we have a concluding discussion in section VII.
Further Background

In 1887, Mendenhall [14] reported one of the earliest known studies in the field of authorship attribution; he used word-length distributions to study certain works of John Stuart Mill, and compare these to works by others on the same topic. In 1901, Mendenhall then applied this method to works of Shakespeare and Bacon [15], however a recent examination of this work concludes [16] finds that the distinctions claimed by Mendenhall were mistaken, revealing distinctions in word-length distributions between poetry and prose, rather than between different author’s styles.

Yule [17] provides the first examination of sentence-length for stylometry, characterising authorship in terms of, for example, mean and standard deviation of number of words per sentence. Seemingly successful in a variety of disputed-authorship cases, sentence-length based studies then became relatively frequent, e.g. [18]. However the more favoured approaches that emerged in the 60s and 70s were those based primarily on function words [1], and on vocabulary distribution. Vocabulary distribution [18, 19] measures the diversity of an author's vocabulary, and tends to involve mathematical models for the frequency distributions of the number of words appearing exactly \( r \) times (for example) for various \( r \). Meanwhile, the use of function words was introduced in [1]; function words (table I) are specifically words that have no significant meaning, but play important grammatical and syntactic roles; they include pronouns, conjunctions, prepositions, auxiliary verbs, and some adverbs. Also, it is known that function words rarely borrow from other languages, and hence it is very rare for new ones to come into fashion, and hence rare for existing functions words to go out of fashion. These various characteristics of function words suggest that the way an individual authors uses them is dependent on his style, rather than affected by confounding factors such as age, era, content, and so on. It is interesting that this class of words are of great value in authorship attribution and similar studies, but are usually directly omitted (i.e. included in the list of stop words – e.g. see [20]) from text mining and related research concerning the content of documents.

As a result of the success achieved by Mosteller and Wallace [1], in using the relative frequencies of function words on the Federalist Papers task, research in their use has flourished, with a variety of classification methods having been studied, but each using function word frequencies as the stylometric ‘fingerprint’. For example, function words were recently used to address the disputed Book of Oz [2], and the disputed Federalist Papers [21, 22].

Given the number of potential function words that can be employed (a ‘primary’ list of 70 is given in [1]), but also given the common scenario in which there are relatively few texts available for which to construct datasets (e.g. in the case of the Federalist papers, there are 66 data points), the authorship attribution problem involves serious challenges regarding overfitting. Consequently, dimension reduction strategies are common in the authorship attribution literature, with principal components analysis (PCA) often employed, along with exhaustive searches of small feature subsets [22]. Support vector machines are also beginning to be used in this field, [6, 22], given their \textit{a priori} expectation of good generalisation performance with relatively sparse data sets.

The Federalist Papers and Book of Oz data

The well-known Federalist Papers are a group of 85 essays, ranging between 900 and 3,500 words in length, all written under the same pseudonym, aimed at persuading the people of New York to approve the U.S. Constitution. 77 of the essays were published in newspapers between 1787 and 1788, and a further 8 were included when they were later published together as a book. The authors of the essays were Alexander Hamilton, James Madison and John Jay; it is known that Hamilton wrote 51 of them, Madison wrote 15 of them, and Jay wrote 5, with a further 3 being co-written by Hamilton and Madison, but a specific set of 11 papers are known as the disputed papers, for which both Madison and Hamilton claimed sole authorship. Hamilton’s essays have a mean length of 2203 words, ranging between 987 and 5,733 words. Madison’s mean number of words is 2755, ranging between 1,704 and 3575 words. Meanwhile the disputed papers average 2022, ranging between 1,133 and 3056. The frequency of the 70 function words is calculated for each individual paper, producing 77 vectors (51 Hamilton, 15 Madison, and 11 disputed). The data are downloadable from [23] and [24].
Two authors, Lyman Frank Baum and Ruth Plumly Thompson wrote the 33 *Adventure Oz* tales. Baum began writing in 1900 and is known to have written at least 14 tales before his death. Then Thompson took over and continued the tales until tale number 33. The 15th *Book of Oz* was published under Baum’s name, one year after his death, and Thompson then claimed that she was the sole author of the 15th Book of Oz.

The 14 tales by Baum and 5 of Thompson’s tales can be easily found in electronic form, from [25] and [26] respectively. The average number of words in Thompson’s tales is 39,017 words, ranging between 33,842 and 45,654 words. The corresponding figures for Baum are 42,100, 38,413 and 53,206. We partitioned each book into several data points by considering pairs of chapters as a unit. In the end, this provided 86 function-word frequency vectors for Baum, 49 for Thompson, and 12 for the single disputed book.

Finally, we note that in both cases, as a result of the wealth of evidence from authorship attribution studies, combined with further historical research, neither the disputed Federalist papers nor the disputed book of Oz is actually disputed anymore. It is generally accepted that the disputed papers are the work of Madison, and the disputed book is the work of Thompson. From the viewpoint of further authorship attribution research studies, this makes both datasets rather less exciting. However, of course it also makes it possible to evaluate and test authorship attribution methods, since we are able to evaluate the accuracy of the results, and consequently these works continue to be used in this research area. We mention finally that the thrust of our own research is, having developed techniques on this and similar test cases, to apply the attribution methods to well-known and historic still-disputed works in the first-authors’ culture.

The Linear Discriminant Analysis Classifier

Given relatively high dimensionality of data compared to the number of data points available in each case, and also given the unbalanced class sizes (51 vs 14 in the Federalist papers case, and 49 vs 12 in the Book of Oz case), we chose to use Linear Discriminant Analysis (LDA) for the classifier (an accessible tutorial is at [27]). LDA naturally and appropriately handles the problems of unequal class sizes. It works simply by finding a linear function of the data vectors that defines a separating hyperplane which separates the data as well as possible, specifically aiming to minimise the ratio of within-class variance to between-class variance. The weights for the discriminating hyperplane are learned by minimizing the cross-entropy error function. Meanwhile, to promote good generalization performance, we use leave-one-out cross-validation, and weight-decay regularization. Weight decay regularization attempts to keep low the absolute values of the weights in the discriminant function, which in turn tends to be associated with better generalization performance, however it involves a parameter which is difficult to choose correctly in advance. We thus repeat the LDA training process several times for different values of this parameter.

Our LDA training implementation uses the Netlab library [28]. The step by step procedure is as follows. We are given a set of vectors to classify (in this paper, all problems are two-class classification problems). Suppose there are \( m \) \( d \)-dimensional vectors (of function-word frequencies) in the training set; we will denote the \( i \)th such vector as \( v_i \), with elements \( v_i,1, v_i,2, \ldots, v_i,d \). An LDA classifier is trained, learning a vector of weights \( w = (w_1, w_2, \ldots, w_d) \) which minimises an error function \( E = E_c + E_d \). This error function is a combination of the cross-entropy error term and the weight-decay regularisation term, where respectively:

\[
E_c = -\sum_{i=1}^{m} t_i \ln(v_i \cdot w) + (1 - t_i) \ln(1 - v_i \cdot w)
\]

and

\[
E_d = \alpha(w \cdot w)
\]

where \( t_i \) is either 0 or 1, denoting the correct class value of vector \( v_i \).

Training is done via the iterative re-weighted least squares algorithm, using default parameters in the Netlab implementation. This training process is repeated for a range of different values of \( \alpha \), estimating the quality for each value by the average performance of the trained LDA using leave-one-out-cross-validation.

Thus, the input to the LDA process is a set of classified training examples, and the output is a classifier, the one corresponding to the (or a) best \( \alpha \) value. The resulting classifier is then used in the following way.
Given a $d$-dimensional test vector, $x$, its classification is indicated by the dot product $x \cdot w$. We convert this into a value between 0 and 1 using the logistic equation, and hence record the value:

$$c_i = \frac{1}{1 + e^{-x_i \cdot w}}$$

for each individual vector $x_i$ of a set of test vectors. This is not vital, but convenient for the next step, which is to compute the ROC curve for the classifier, by calculating the false positives and true positives ratios on the test set for each of several threshold values between 0 and 1. In other words, the output of the classifier is a number between 0 and 1, where the expectation is (for example, where just two authors are involved) that texts written by one author will lead to an output closer to 0, and texts written by another will lead to an output closer to 1. Any particular threshold $t$ between 0 and 1 will lead to a point on the ROC curve. For example, setting the threshold at 0.3 indicates that we class test inputs as being written by author $A$ if the output is below 0.3, and by author $B$ if the output is above 0.3. This leads to a specific pair of points on the curve (proportion of correctly classified author $A$ texts plotted against proportion of correctly classified author $B$ texts). A collection of thresholds therefore leads to a curve. One measure of the classifier’s performance is the area under this curve. An area of 1 indicates perfect performance.

**The Hybrid ROC-dominance based Approach**

In addition to straightforward principal components analysis (PCA), we report here the investigation of two algorithms that hybridise simple evolutionary algorithms with the LDA training process described above. In this section we describe the first of these, which wraps a simple evolutionary algorithm around the LDA process, but using an idea from [29] as a non-standard way to select parents, based on using an archive of non-dominated ROC curves.

The basic procedure is as follows. Each chromosome encodes a non-empty (but otherwise unrestricted) subset of the 70 primary function words from [1]. The encoding used is simply a list of features. The fitness of a chromosome in this case is its ROC curve – hence this a multiobjective approach [30—32] (although we do not yet employ any sophisticated or up-to-date strategies from the multiobjective evolutionary algorithm literature in this work). This is obtained by running the LDA process described above on the training data to produce the best classifier, and generating the ROC curve from that classifier.

Initially, we start with a population of one, which represents a subset of size one – i.e. the initial chromosome encodes a singleton, representing a randomly chosen one of the 70 function words. More generally, while the algorithm is running, there is an archive of chromosomes maintained, which is mutually nondominated w.r.t. their ROC curves. No size limit is enforced for this archive (see discussion about such issues in [33]).

The algorithm proceeds as follows, following the generation, evaluation and archiving of the initial random solution. For $gen$ iterations, we select a chromosome from the archive, and then randomly choose to either add, delete, or change a randomly chosen feature (naturally, only valid choices are made, so that we do not delete a feature from a singleton set, or duplicate an existing feature, etc.). The resultant mutant is evaluated, and then the archive is appropriately updated.

**The Hybrid AUC-Fitness Approach**

Our second approach also wraps a simple EA around the LDA training process, but this times simply calculates the area under the ROC curve (AUC), and treats this as a single-objective fitness value to be maximised. For this approach we use a straightforward small-population steady-state evolutionary algorithm. Specifically, population size 5, mutation (only) operator as described in section V, binary tournament selection, and replace-worst replacement, breaking ties by number of features. That is, in each generation, binary tournament selection is used to choose a parent. A mutant is then generated and evaluated. The mutant enters the population if it is at least as fit as the current worst. If there is a tie between the mutant fitness and the fitness of the current worst, but the mutant contains more features than the current worst, then the mutant is discarded.

As with the ROC-dominance approach, the initial population contains only randomly chosen singleton feature vectors.
Experiments and Results

For each of the Federalist Papers and Book of Oz in turn, we will now present the results of preliminary analysis with PCA, and then the two hybrid approaches.

Federalist Papers: Principal Components Analysis

Following standard PCA applied to the Federalist Papers data, we plotted the projections of the data onto the first two principal components in Figure 1. It can be seen that the general positions of the eleven disputed papers (squares) are not able to be clearly distinguished from either the Madison papers (crosses) or the Hamilton papers (squares). Arguably, they are shifted more towards the ‘Madison space’ than the ‘Hamilton space’, but several individuals are much closer to Hamilton papers than to any Madison paper.

Following PCA, we used the principal component transformations of the data as input to the LDA process already described, trying this for the first $k$ principal components, for each $k$ from 2 to 70. For each such $k$, we measured validation error by recording the cross-entropy error on the disputed papers. Figure 2 shows the plot of validation error against number of principal components used. Clearly, the findings from PCA are that we need the first 15 principal components to reach the minimum error, tentatively suggesting that around that many function words may be needed (in the sense that this is the suggested number of latent features required for good performance).

We ran each of the Hybrid ROC-dominance approach of section V (ROCD) and the Hybrid AUC-fitness approach (AUCF) 10 times with the following parameters. We performed 300-iteration runs, where each evaluation incorporated 5 runs of the LDA training process for different values of $\alpha$ randomly chosen between 0.1 and 1. AUCF used a population size of 5. Results are summarized in table II. The table reports results on the training set, but not on the disputed papers. On the disputed papers, the best of 10 trials for 300 iterations was a set of 3 function words.

Figure 1: Projection of the first two principal components of the Federalist Papers data.
Appendix B: Publications

Figure 2: Error on the disputed papers against number of principal components learned from the test data: Federalist papers.

**Federalist Papers: The Hybrid ROC-Dominance Method and the Hybrid AUC-Fitness Approach**

There is insufficient evidence so far to support a statistical claim that AUCF is a better approach than HROC on this problem, however the important and interesting findings are that these methods can both reliably obtain classifiers that use only 2 function words. In the case of AUCF, the LDA classifiers associated with each best-fitness set of words emerging from the training process was tested six times on the training data with randomly perturbed values of the regularization parameter. When the training set classification was perfect each time, this set of function words (and its associated classifier) was regarded as *stable*, and used on the unseen test set of disputed papers. In all cases, such a stable set of function words also achieved perfect discrimination on the disputed papers. None of the 2-function word sets was stable, however the best of 10 trials at 300 cycles found a stable set of 3 words. Unfortunately, at the time of writing we have not been able to collect the corresponding test results for HROC.

<table>
<thead>
<tr>
<th>Table II: HROC/AUCF methods training results on FED. papers.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
<tr>
<td>-----------------</td>
</tr>
<tr>
<td>Best of 10 trials after 100 cycles</td>
</tr>
<tr>
<td>Mean of 10 trials after 100 cycles</td>
</tr>
<tr>
<td>Best of 10 trials after 300 cycles</td>
</tr>
<tr>
<td>Mean of 10 trials after 300 cycles</td>
</tr>
</tbody>
</table>

In comparison, Fung [6], using support vector machines, found a classifier that also used 3 function words (*to*, *upon*, and *would*), while Bosch and Smith [22] achieved the same result with an extensive test that searched all combinations of 1, 2 and 3 function words using a linear programming...
formulation, discovering a single set of 3 (as, our and upon) that achieved perfect classification. In our case, HROC was able to find a classifier that worked only with such, upon and with, while different perfectly-classifying sets of words were found by AUCF, but usually including the word upon.

**The Book of Oz: Principal Components Analysis**

Figures 3 and 4 respectively show the projections of the first two principal components for the Book of Oz data, as follows. In figure 3, each tale is a separate point, while in figure 4 we plot the 86 points obtained by dividing the tales into two-chapter chunks. ‘Thompson space’ is represented by the crosses, Baum is represented by circles, and the disputed book is represented by squares.

Again, it is not clear from PCA alone to whom authorship of the disputed book should be attributed. These plots do seem to lean a little towards Thompson, however far from enough to have any real confidence in that conclusion. When using the principal components vectors in tandem with the LDA classifier, we found that the best validation error was obtained when using the first 26 principal components (compare with 15 for the Federalist papers). This provides evidence that the attribution task for the Book of Oz is more complicated than for the Federalist Papers, needing correspondingly more features to obtain a good distinction between the two authors’ writing styles. The corresponding plot of validation error against number of components is in Figure 5.

**Book of Oz: HROC and AUCF**

We ran each of the Hybrid ROC-dominance approach of section V (ROCD) and the Hybrid AUC-fitness approach (AUCF) 10 times with the same parameters as described in section VII.B. Training results are summarized in Table III; meanwhile, using the same approach to choosing sets of words for analysis of the (unseen) disputed works, we again found that stable best-performing classifiers from the AUCF training runs always produced perfect results on the test set. The best of the 10 runs at 300 iterations found a stable set of 6 words.

Again, there is insufficient evidence so far to support a statistical claim that AUCF is a better approach than HROC on this problem, however we again have interesting findings that show that each method is adept at reliably discovering relatively small subsets of features that can perform perfected discrimination of the disputed work. Previous work on this case is less common than in the case of the Federalist papers, and we do not have comparable results concerning attempts to minimise the number of features. Binongo [2], concentrated on using principal components of a set of 50 function words. We feel it is therefore an interesting contribution that we have found sets of six words that can lead to perfect discrimination on the (unseen) disputed works.

**Figure 3**: Projection of the first two principal components of the Book of Oz data – one point per tale.
Figure 4: Projection of the first two principal components of the Book of Oz data – one point per pair of chapters.

Figure 5: Error on the disputed papers against number of principal components learned from the test data: Book of Oz.

<table>
<thead>
<tr>
<th>Table III: HROC/ AUCF methods training results on Book of Oz.</th>
</tr>
</thead>
<tbody>
<tr>
<td>HROC</td>
</tr>
<tr>
<td>Best of 10 trials after 100 cycles</td>
</tr>
<tr>
<td>Mean of 10 trials after 100 cycles</td>
</tr>
<tr>
<td>Best of 10 trials after 300 cycles</td>
</tr>
<tr>
<td>Mean of 10 trials after 300 cycles</td>
</tr>
</tbody>
</table>
Concluding Discussion

In conclusion, we addressed the problem of Authorship Attribution using hybrids of simple evolutionary search and a linear discriminant classifier, using evolutionary search to find small function word subsets as the sets of features used in training the classifier. Baseline comparison was also done with using straightforward PCA to transform the data, finding that 15 and 26 components were needed respectively to obtain perfect performance on the disputed Federalist Papers and Book of Oz respectively. Using a simple EA to search feature subsets based on iteratively selecting randomly from subsets with so-far nondominated ROC curves (the HROC approach), we were able to reliably find subsets of function words of sizes 3 and 6 respectively. Regarding the Federalist Papers task, this equals what has been achieved before in the literature, which in turn has some implications and interest for stylometry studies. In the case of the much more difficult Book of Oz case, we can only conclude that the result seems very good given the large number of principal components required, and perhaps sets a target for related studies, since work so far has not used the Book of Oz task in an explicit attempt to minimise the number of function words used for discrimination. A simple EA for evolving ROC curves, again hybridised with the LDA classifier (which we called AUCF), achieved slightly better results here during training than the HROC approach.

Our work on this so far has been hampered by the long training times required by the LDA classifier built into our fitness function, and the corresponding repeated runs of that process that are required to find a good parameter for the weight decay regularisation. In ongoing work we will compare this with less time-consuming classifiers, and so it has yet to be seen whether similar or better results can be achieved with a less sophisticated classifier. Finally, it is clear that evolutionary algorithms have a potential role in authorship attribution, and stylometry in general, particularly regarding feature selection.

REFERENCES

Appendix B: Publications

[26] http://onlineBooks.library.upenn.edu
[27] http://marketing.byu.edu/htmlpages/tutorials/discriminant.htm
Authorship Attribution in Arabic using a Hybrid of Evolutionary Search and Linear Discriminant Analysis

Kareem Shaker and David Corne
School of Mathematical and Computer Sciences, Heriot-Watt University
Edinburgh, EH14 4AS, UK
ks113@hw.ac.uk, d.w.corne@hw.ac.uk

Abstract

Authorship Attribution is the problem of determining the authorship of one or more texts. Applications include disputed authorship, or deciding which of a collection of pieces of text were by the same author. A popular and successful approach is to characterize a specific author in terms of the usage pattern of function words. These are common words that are unrelated to subject matter, and tend to be used in specific ways by different authors. In English, a well-known collection of 70 function words is often used for this purpose. Previously, using a hybrid of evolutionary search and linear-discriminant analysis (LDA), we have shown excellent performance in authorship attribution in English based on a function word approach. Here, for the first time, we propose and test a set of Arabic function words for use in Arabic authorship attribution. Tests indicate that the chosen collection forms an effective basis for authorship attribution in Arabic.

1. Introduction

The Authorship Attribution problem is the task of determining the authorship of a given piece of text. In cases of disputed authorship, two (or maybe more) distinct
individuals may claim authorship, and there are several historical examples of such conflicting authorship claims. For example, two well-known cases of disputed texts in English include the disputed Federalist papers [1] and the 15th Book of Oz [2].

A wide variety of methods have been researched for authorship attribution (e.g. see [3] for a survey). The main issue of interest is how to represent an author’s ‘fingerprint’, which overlaps almost completely with the issue of how to encode a piece of text as a feature vector. A subsidiary issue is the choice of machine learning method that will then be used to produce classifiers, that will in turn attempt to predict authorship for disputed stretches of text. As yet there is no clear convergence on any particular encoding or machine learning approaches, but a certain approach to the encoding of text is particularly popular and successful; this relates to the use of function words.

Table I: Mosteller and Wallace [1] function words.

<table>
<thead>
<tr>
<th></th>
<th>a</th>
<th>all</th>
<th>also</th>
<th>an</th>
<th>and</th>
</tr>
</thead>
<tbody>
<tr>
<td>any</td>
<td>are</td>
<td>as</td>
<td>at</td>
<td>be</td>
<td></td>
</tr>
<tr>
<td>been</td>
<td>but</td>
<td>by</td>
<td>can</td>
<td>do</td>
<td></td>
</tr>
<tr>
<td>down</td>
<td>even</td>
<td>every</td>
<td>for</td>
<td>from</td>
<td></td>
</tr>
<tr>
<td>had</td>
<td>has</td>
<td>have</td>
<td>her</td>
<td>his</td>
<td></td>
</tr>
<tr>
<td>if</td>
<td>in</td>
<td>into</td>
<td>is</td>
<td>it</td>
<td></td>
</tr>
<tr>
<td>its</td>
<td>may</td>
<td>more</td>
<td>must</td>
<td>my</td>
<td></td>
</tr>
<tr>
<td>no</td>
<td>not</td>
<td>now</td>
<td>of</td>
<td>on</td>
<td></td>
</tr>
<tr>
<td>one</td>
<td>only</td>
<td>or</td>
<td>our</td>
<td>shall</td>
<td></td>
</tr>
<tr>
<td>should</td>
<td>so</td>
<td>some</td>
<td>such</td>
<td>than</td>
<td></td>
</tr>
<tr>
<td>that</td>
<td>the</td>
<td>their</td>
<td>then</td>
<td>there</td>
<td></td>
</tr>
<tr>
<td>thing</td>
<td>this</td>
<td>to</td>
<td>up</td>
<td>upon</td>
<td></td>
</tr>
<tr>
<td>was</td>
<td>were</td>
<td>what</td>
<td>when</td>
<td>which</td>
<td></td>
</tr>
<tr>
<td>who</td>
<td>will</td>
<td>with</td>
<td>would</td>
<td>your</td>
<td></td>
</tr>
</tbody>
</table>
The use of so-called *function words* for authorship attribution was introduced by Mosteller and Wallace [1]. The idea is that an author's style can be characterised in terms of the frequencies with which that author uses each of a relatively small number of specific words. These are ‘function words’ in the sense that their use should be independent of the content or subject matter of any given text. E.g. if a writer writes one essay about cars, and another essay about flowers, these two essays will show quite different overall distributions of words, however when only the distributions of the *function words* are considered, we might expect no significant differences for these two essays, given that they were written by the same author. However, the hypothesis behind function words is that there may be significant differences in the function word usage between different authors. This hypothesis has been borne out in several studies [4]. The primary list of 70 function words used by Mosteller and Wallace [1] is provided in Table I.

Typically, given an authorship attribution or related task, researchers will choose a set of function words, and construct datasets consisting of frequency vectors of function word usage, for each of several sections of texts with known authorship. A statistical and/or a machine learning method is then applied to these data, yielding a classifier. The classifier is then applied to test data, which, in a real case of disputed authorship, will be function word frequency vectors associated with the disputed text(s). The familiar range of classification methods have been attempted, including neural networks [5], support vector machines [6], and various statistical and probabilistic approaches [7, 8, 9], including linear discriminant analysis [17, 19] and evolutionary search [11, 12, 13, 17].

In recent work [17] we explored a hybrid of evolutionary search with linear discriminant analysis for authorship attribution in English, where the emphasis was on attempting to find minimal stylistic fingerprints (i.e. a small set of function words) that were sufficient for the cases studied. In that work, we evolved feature subsets for classification by linear discriminant analysis (LDA), using the area under the ROC-curve (Receiver Operator Characteristic) as our fitness measure following the training of the classifier. This hybrid EA/LDA approach, which we use here, involves steps to ensure good generalisation performance in the parameterisation of the LDA, and found excellent results for the celebrated disputed authorship (in English) cases that were studied (the Federalist papers and the Book of Oz). The approach was particularly good at finding minimal subsets of English function words that could support accurate classification; this is of particular interest in the general study of stylometry [10]. However in this paper we are only concerned with predictive accuracy.

Finally, we note that there has been very little study of function words in alternative languages, and certainly no studies can be found that attempt to posit and test function words for authorship attribution in Arabic. In this paper we introduce and test function words in Arabic, motivated in part by a number of disputed authorship scenarios in the Arabic religious literature (although it is proving hard work to obtain the associated texts in electronic form). There are clearly other applications for an Arabic function word set, including (as with any language) questions of stylistic analysis, plagiarism investigations, and other investigations.

The remainder is set out as follows. In section 2 we provide further background on Authorship Attribution and function words, while in section 3 we expand on our hybrid EA/LDA classifier. Section 4 describes our dataset of Arabic novels, and section
5 reports experiments and results that refined and tested sets of Arabic function words. We summarise and discuss in section 6.

2. Background and Related Work

Authorship attribution studies began in 1887, when Mendenhall [14] reported using word-length distributions to study certain of the works of John Stuart Mill, comparing them to work by others on the same topic. Mendenhall followed this up in 1901, by applying his method to certain works of Shakespeare and of Bacon [15]. Though seminal, Mendenhall’s work can be criticised [16] for mistakenly revealing differences in word-length distributions between poetry and prose, rather than between different author’s styles.

The first to examine sentence-length distribution (rather than word-length) was Yule [17], who attempted to characterise authorship in terms of, for example, mean and standard deviation of number of words per sentence. This method showed some success, leading to more sentence-length based studies [18], however such studies were supplanted in the 1960s by characterisations in terms of function words [1], and more generally on vocabulary distribution [18, 19], which measures the diversity of an author's vocabulary. Typically, vocabulary distribution is modelled by frequency distributions of the number of words appearing exactly \( r \) times (for example) for various \( r \). The function words approach [1], in contrast, deals only with specific words (such as pronouns, conjunctions, prepositions, and so forth) that have no significant meaning, but are grammatically and syntactically important.

Research in function word based authorship attribution flourished (e.g. [2, 21, 22]) following Mosteller and Wallace’s demonstration of their use for the case of the disputed Federalist Papers [1]. Work in this area still tends overwhelmingly to concern English texts, and focuses on the comparison of different learning methods and/or augmentations to a function word approach [e.g. 6, 22].
3. The EA/LDA Classifier

One of the main challenges for machine learning approaches in authorship attribution is the typically small size of datasets in terms of number of samples (frequency vectors). Methods that incorporate careful handling of over-fitting are therefore common, such as linear discriminant analysis [27] and support vector machines. In this article we use the hybrid Evolutionary Algorithm / Linear Discriminant Analysis classifier (EA/LDA) described in [17], using the variant that evaluates fitness using the area under the ROC curve returned by the LDA classifier. The role of the EA is to find a subset of the function words that are in turn used to train the LDA. The LDA works simply by finding a linear function of the frequency vectors that defines a hyperplane which separates the data as well as possible, minimising the ratio of within-class variance to between-class variance. The weights for the discriminating hyperplane are learned by minimizing the cross-entropy error function.

In straightforward terms, the role of the LDA classifier is as follows. First, a subset of function words is supplied by the EA (i.e. a chromosome defines a subset of the function words). The input to the LDA is then the set of labelled training vectors, reduced in dimension (i.e. retaining only the elements indicated in the EA chromosome). The LDA then learns a weight vector, characterising a good separating hyperplane for the two classes (authors). The weight vector is then used for classification simply via considering its dot product with a test vector. The dot products are transformed between 0 and 1 by the logistic equation, and this value essentially represents a fuzzy decision (with one author associated with 0, and the other associated with 1). Following consideration of all test inputs (leave-one-out cross-validation is used), by considering a series of threshold values, an ROC curve is then constructed (equivalently, the curve indicating the tradeoff profile between false positives and true negatives).

Further detail is given in [17], while an accessible reference to LDA is in [27], and our LDA uses the Netlab library [28]. Finally, we describe aspects of the Evolutionary Algorithm (EA) used. A simple EA is wrapped around the LDA training process; as indicated, the EA supplies the chromosome (a subset of the available function words) and the LDA evaluates it, supplying in the end an ROC curve summarizing performance on the accumulated validation cases during leave-one-out cross-validation. The EA simply uses the area under this curve (AUC) as the fitness value to be maximised. As in [17], the EA is otherwise a straightforward steady-state evolutionary algorithm with a population size of 5, a mutation (only) operator that, with equal probability, either deletes a random feature, changes a feature, or includes a new feature, binary tournament selection, and replace-worst replacement, breaking ties by number of features (preferring fewer). In each generation, a parent is chosen via binary tournament selection, a mutant is then generated and evaluated, and then enters the population if it is no worse than the current worst. In the case of a tie between mutant and current worst fitness, the mutant is retained only if it does not contain more features than the current worst. Finally, the initial population contains only randomly chosen singleton feature vectors.
4. Arabic Text Datasets

To derive and test a collection of Arabic function words we procured a dataset of 14 books by six different writers. These were obtained from the website of the Arab Writers Union (www.amu-dam.net). The books ranged in size from 13,987 words to 37,567 words, with a mean of 23,942 words. Each of these books (details in Table II) were downloaded and processed to convert them into a string of Arabic words without extraneous characters and spaces.

5. Experiments: Refining the Arabic Function-Word Set

The initial set of Arabic function words was based on creating a collection of common prepositions and conjunctions, mirroring the semantic structure of the Mosteller and Wallace set for English. This led to a collection of 106 Arabic words.

We then investigated the frequencies of each of these 104 words over the complete collection of 14 books. This revealed that around 40 of the words were particularly common among all of the books, with patterns of usage that (on first sight) appeared roughly uniform, while a further 40 of these words tended to be of particularly low frequency. In preliminary work not reported here in detail, we used the test cases (Table III) to evaluate two subsets of words in turn. First, a collection of 64 words (omitting only the most frequent ones), and secondly a collection of 65 words (omitting only the least frequent). Only the latter set was found to be particularly promising, and we include results from this collection of 65 words below. Meanwhile, this set (which we call AFW65) is given in Table IV, which gives a numeric ID, the Arabic representation, and (in most cases) a ‘ballpark’ translation into English.
Table II: Details of the Arabic books dataset; includes shortened IDs for books and authors used later in presentations of results.

<table>
<thead>
<tr>
<th>Authors</th>
<th>Book details and IDs</th>
</tr>
</thead>
</table>
| Ibrahim Khalil | Haris Al Maiz  
(HAM: 14,679 words) |
| | Sodom Sebake Al Awez  
(SSAA: 2 parts, 28156 words and 29518 words) |
| Basem Ibrahim Abdo | Gesr Al Mawt  
(GAM, 37567 words) |
| | Zahra fi Al Remal  
(ZFAR: 2 parts, 23241 words and 26061 words) |
| Taleb Omran | Ahzan Al Sinbad  
(AAS, 20389 words) |
| | AlBood Al Khamis  
(AAK, 13987 words) |
| | Madina Kharig Al Zaman  
(MKAZ, 14513 words) |
| | Al Fetiah Al Aghrar we Asfâr al Kashf  
(AFAA, 19063 words) |
| Mary Show | Defly (DE, 24062 words) |
| | Awel Hob and Akheir Hob  
(AHAH: 2 parts, 18848 words and 19807 words) |
| Mohamed Youssef Salibi | Al Taih (AT, 36892 words) |
| | Sebahaa fi Al Wahl  
(SFAW: 2 parts, 25647 words and 28274 words) |
| Hessen Abd Al Kareem | Al Nabaa (AN: 2 Parts 29644 and 29472 words) |
| | Shagaret al Toot (SAT: 2 parts, 19024 words and 19998 words) |
Table III: Details of the five test cases used in experiments; in each case, books from two authors constitute the training set, and different books from the same two authors comprise the test set.

<table>
<thead>
<tr>
<th>Test case</th>
<th>Train and test set details</th>
</tr>
</thead>
</table>
| A         | Training set: Books AAK and HAM  
|           | Test set: Books MKAZ and SSAA |
| B         | Training set: Books AHAH and SATP1  
|           | Test set: Books DE and ANP1     |
| C         | Training set: Books GAM and SFAWP2  
|           | Test set: Books ZFARP1 and AT    |
| D         | Training set: Books ANP1 and AAS  
|           | Test set: Books ANP1 and AFAA    |
| E         | Training set: Books AFAA and ZFARP2  
|           | Test set: Books MKAZ and GAM     |

Table IV: AFW65 – a set of 65 Arabic function words, constructed by positing a collection of 104 candidate words, and removing those with low frequency in a collection of Arabic books

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>في</td>
<td>من</td>
<td>على</td>
<td>إلى</td>
</tr>
<tr>
<td>in</td>
<td>From</td>
<td>about</td>
<td>over</td>
<td>to</td>
</tr>
<tr>
<td>6</td>
<td>حتى</td>
<td>فلا</td>
<td>منذ</td>
<td>لا</td>
</tr>
<tr>
<td>till</td>
<td>Not</td>
<td>since</td>
<td>no</td>
<td>then</td>
</tr>
<tr>
<td>11</td>
<td>بل</td>
<td>12</td>
<td>أو</td>
<td>13</td>
</tr>
<tr>
<td>14</td>
<td>أن</td>
<td>15</td>
<td>19</td>
<td>20</td>
</tr>
<tr>
<td>16</td>
<td>كان</td>
<td>17</td>
<td>إن</td>
<td>إن</td>
</tr>
<tr>
<td>18</td>
<td>إذن</td>
<td>19</td>
<td>كي</td>
<td>20</td>
</tr>
<tr>
<td>20</td>
<td>as if</td>
<td>The</td>
<td>theref</td>
<td>So</td>
</tr>
</tbody>
</table>
### Appendix B: Publications

<table>
<thead>
<tr>
<th>No.</th>
<th>Arabic</th>
<th>English</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>لا 20 21 ل 22 ما 23 أي 24 أما 25 ألا</td>
<td>no What Any Not</td>
</tr>
<tr>
<td>2</td>
<td>إذا 26 ها 27 إذ 28 إذا 29 لو 30 لولا</td>
<td>If If</td>
</tr>
<tr>
<td>3</td>
<td>بي 31 بل 32 يا 33 تعم 34 بلا 35 هذا 36 بل</td>
<td>Is? Oh Yes without this</td>
</tr>
<tr>
<td>4</td>
<td>37 هذه 38 ذلك 39 هؤلاء 40 أولئك</td>
<td>that this Such they those</td>
</tr>
<tr>
<td>5</td>
<td>41 الذي 42 التي 43 الذين 44 هو 45 هم</td>
<td>who whom which Whose he them</td>
</tr>
<tr>
<td>6</td>
<td>46 هي 47 أنت 48 أنتم 49 أنا 50 نحن</td>
<td>she You you you are I we</td>
</tr>
<tr>
<td>7</td>
<td>51 الآن 52 بين 53 هنا 54 هناك 55 كان</td>
<td>now between Here there been</td>
</tr>
<tr>
<td>8</td>
<td>ليس 56 نيس 57 أصبح 58 ظل 59 مما 60 لماذا</td>
<td>not became Keep what why</td>
</tr>
<tr>
<td>9</td>
<td>كيف 61 كم 62 كم 63 أين 64 إلى 65 مهما</td>
<td>how how how many Where when what ever</td>
</tr>
</tbody>
</table>
At this point, some preliminary notes about experimental setup is in order. The common way in which a set of function words is employed (and which we do here) is to transform a section of text (a ‘chunk’) into a vector of $n$ numbers in the interval $[0, 1]$, each indicating the frequency of a function word as a proportion of the total words in that chunk. That is, if the chunk of text contains 1,000 words, and element $i$ of this vector is 0.022, this indicates that function word $i$ occurs 22 times in that chunk. To formulate an authorship attribution problem (or simply a simulated such problem) as a data mining task, a large section of text, such as a book, is partitioned into chunks of $c$ words, for some $c$, and a frequency vector is built for each chunk. Each such frequency vector then is then associated with a target class, which in turn is simply the author of that chunk.

Hence, in each test case, an authorship dispute is simulated by supposing that we have two authors, Author1 and Author2, who both claim to have written each of Book1 and Book2. The training set comprises an undisputed book from each of the two authors, while Book1 and Book2 comprise the test set. In test case A, for example (see Table III), the training set comprises XXX chunks from the book ABC and YYY chunks from the book BCD, and the test set comprises XX chunks from the book CDE and YY chunks from the book DEF.

An important consideration is the size of the chunks. A feel for this can be gained from considering the extremes. If the chunk sizes were very small, function word frequencies would generally be very low, and often zero, and we would expect that each chunk would be too small to capture the stylistic fingerprint of an author. If the chunk sizes were very large (e.g. we could transform an entire book into a single frequency vector), we would have too few samples to do reliable machine learning, and could expect poor generalization performance.

A number of preliminary experiments were done with different chunk sizes, and we report here the more successful sizes, which were 1,000 and 2,000 words respectively. Table V summarises results on the five test cases for the function word set AFW65, for each of 1,000 and 2,000 word chunks. Each entry in the table corresponds to the mean of five trial runs, each of which ran for a specific parameterization of the hybrid classifier (determined in advance from preliminary experiments, and similar to the configuration that achieved best results for English authorship attribution in [17]). The result of an experiment is a percentage accuracy figure, which indicates the
percentage of chunks in the test set that were correctly labeled by the classifier; this figure is always the mean over 5 trials. Notice that, in one sense, the accuracy figures understate the potential performance of set AFW65 for authorship attribution. If the authorship of a book is disputed, and the ‘decision’ of the experiment was made on the basis of the author to whom most chunks were attributed, then, every experiment reported below would yield the correct result (this was not the case in preliminary experiments with a full set of 104 words, or with a prior set which removed the most common words). In interpreting these results, higher accuracy therefore tends to indicate better reliability – i.e. the degree to which we might expect a correct attribution when only a relatively small number of words are available in the disputed text.

Table V: Summary results of experiments on the five test cases using function word set AFW65 (Table IV).

<table>
<thead>
<tr>
<th>Test case</th>
<th>1,000 word chunks</th>
<th>2,000 word</th>
<th>3000 word</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>AFW65</td>
<td>AFW54</td>
<td>AFW65</td>
</tr>
<tr>
<td>A</td>
<td>76.14%</td>
<td>87.61</td>
<td>87.74%</td>
</tr>
<tr>
<td>B</td>
<td>90.12%</td>
<td>90.49</td>
<td>89.50%</td>
</tr>
<tr>
<td>C</td>
<td>80.19%</td>
<td>78.83</td>
<td>82.13%</td>
</tr>
<tr>
<td>D</td>
<td>86.44%</td>
<td>86.98</td>
<td>84.57%</td>
</tr>
<tr>
<td>E</td>
<td>89.83%</td>
<td>84.60</td>
<td>88.38%</td>
</tr>
</tbody>
</table>

To further refine the set of Arabic function words, we examined the occurrences of each of the 65 words on AFW65 in the dataset, and considered the variance of their frequencies across the set of 2,000 word chunks from different authors. That is, if a function word has a low variance across chunks for different authors, then different
Appendix B: Publications

authors tend to use that word with the same frequency, and it may not contribute materially to authorship attribution studies. We found 11 such words with relatively low variance, and composed the function word set AFW54 by eliminating these words. AFW54 comprises the set shown in Table IV, with the following removed: 16, 18, 30, 39, 40, 45, 48, 58, 63, 64, 65. Table V also shows the corresponding results on the five test cases when using the new set AFW54.

As mentioned previously, each trial of each experiment for each test case was able to accurately predict the authorship of each of the test books, in the case that we regard the authorship attribution decision as the majority vote of predicted authorship of a book’s chunks. In finer detail, the accuracy results (percentage of chunks with accurately predicted authorship) indicate the reliability of the underlying method, which is of particular interest when there is a need to attribute the authorship of a relatively small test body of text. It is not straightforward to compare these accuracy figures with other authorship attribution studies, but we report that they compare very favourably with accuracies reported, for example in [18] for Greek texts using a variety of methods, and in [19] for Dutch texts using a linear discriminate classifier. Finally, given the number of trial runs and case studies, we cannot indicate any statistically significant difference between function word sets AFW65 and AFW54, however each is statistically superior to the original complete set of 104 words.

6. Summary, Discussion and Conclusion

We introduced the use of Arabic function words for use in Arabic authorship attribution and related studies for Arabic texts. Our starting point for a set of Arabic function words was based on collection of 104 common function words reflecting the semantics of the English function words from Mosteller and Wallace [1]. Following a collection of experiments and analyses, using a dataset of Arabic novels, we have refined this to two sets of words AFW65 (Table IV), and AFW54 (Table IV, with 11 words omitted as detailed in section 6). Each of AFW65 and AFW54 was used as the basis to transform a number of Arabic texts into frequency vectors, and the ‘performance’ of these word sets was assessed by experiments that used a hybrid of an EA and LDA to produce a classifier, and then tested that classifier on unseen data. The resulting performance was clearly in line with results that have been noted for authorship attribution studies in other languages. Set AFW54 is arguably a better choice, however we cannot make that claim with any statistical significance. For the cases considered here, only limited
investigation is reported for assessing the appropriate ‘chunk’ size. For real applications this will likely depend on several factors, but we have determined (partly from preliminary experiments with smaller chunk sizes) that at least around 1,000-word chunks are necessary to obtain adequate characterization of function word usage for Arabic authors.

Arguably, this work has confirmed that the concept of function words translates suitably well into the Arabic language. In other words, different authors, by and large, use this set of words in sufficiently different ways, enabling us to capture the stylistic fingerprints of individual authors and use these to distinguish between authors.

References


