**Stylometry and authorship attribution**

In this paragraph we will situate our methodology in the field of stylometry and discuss how it fits in current trends in this domain. The present-day field of ‘stylometry’ is commonly defined as the quantitative study of writing style (Holmes 1998). Stylometry is an important subfield in the Digital Humanities (Schreibman et al.), but also attracts many contributions from more technical research domains outside the Humanities, such as Computational Linguistics, Information Retrieval and Machine Learning. Authorship studies are probably the most popular application of stylometry currently (Juola, Stamatatos, Koppel et al.), with a rich history ranging back to the famous, foundational monograph on the *Federalist papers* by Mosteller and Wallace. In authorship studies, researchers attempt to establish a quantitative link between an author’s identity and his/her unique writing style or stylistic ‘fingerprint’. Thus, stylometry is broadly speaking underpinned by the – not uncontroversial – assumption that each author adopts a relatively unique writing style or ‘stylome’ (Van Halteren et al.), which can captured using computational models.

The field of computational authorship studies underwent an interesting evolution in recent decades, which can be characterized by an interesting interplay between, on the one hand, methodology-oriented research in more technical fields and, on the other hand, case-oriented studies in e.g. literary studies (Love, 2002). The main novelty which was introduced by Mosteller and Wallace is that they proposed to model writing style on the basis of function words in texts, while attribution studies until then had largely relied on the analysis of content words (on competing notions of style, see Van Dalen et al.). This idea has been further developed in John Burrows’s seminal monograph on the novels of Jane Austen. Function words are indeed a very attractive category of stylistic enquiry (Kestemont 2014). From a statistical perspective, function words prove to be very robust features, in the sense that they occur frequently throughout all texts and thus represent variables which most authors share. Thus, they are much less ‘sparse’ variables than content words. Additionally, they offer the advantage that they are less tied to specific topics or genres than content words (Kestemont et al.). Psycholinguistic evidence moreover suggests that humans process function words *unconsciously* to a large extent; there ‘inconspicuous’ nature might render function words less vulnerable to forgery, or other forms of (malignant) stylistic imitation.

Nevertheless, function words also have a number of issues. Firstly, some functions words do display strong correlations with genre. Narrative perspective in novels is a good example in this respect, since the adoption of a first-person or third-person narrator will have a decisive influence on the frequencies of personal pronouns in texts. Secondly, it has noted that techniques based on function *words* are much less suited for the analysis of more highly-inflected languages such as Finnish or Polish (Rybicki & Eder). Therefore, the field increasingly moves to feature types which can also take into account morphemic information at the sub-word level. The kind of ‘character’ n-grams which we will discuss in more detail below are an excellent example of a feature type which is more apt at this. Thirdly, while the analysis of function words has proven extremely successful for longer (e.g. novel-length) texts, the field is increasingly moving to harder problems, involving the analysis of much shorter (e.g. poem-length) texts. In shorter texts, even function words become relatively sparse variables and there is therefore an understandable tendency to include as much information as possible, also from content words.

An other interesting evolution in stylometry involves the transition in experimental setup from closed *attribution* problems, to more open, *verification* problems. Traditional studies in authorship identification formalized the problem as a traditional classification task, where an algorithm would be trained to attribute an anonymous texts to one of series of candidate authors, for which training material was available in the form of writing samples of which the authorship could be ascertained. The setup resembles the situation of a police line-up; crucially, the attribution algorithm could safely assume that the correct author was included in the available target authors. In this setup, strong results have been obtained in empirical studies, especially in cases where longer texts where available and the number of target authors was relatively small (Luyckx & Daelemans). It should be stressed that the closed attribution case is not entirely unrealistic: in many real-world forensic case studies, it can indeed be ascertained that a particular text (e.g. bomb letter) must have been written by one of series of suspects. Nevertheless, there are infinitely more situations in which it impossible to guarantee to that the actual author of an anonymous is included in the list of candidate authors for which we have training data available.

In recent years, the field has therefore increasingly focused on the problem of authorship verification, in which it is not assumed that the author of an anonymous is by necessity included in the candidate authors. Thus, this setup adds the ‘None of the above’ option to the classification problems. Naturally, this makes the verification setup much more difficult than the standard attribution approach. Koppel et al. have reported encouraging results for the verification setup in recent years. In a recent paper, they even took the verification problem one step further: given any text of pair of a source text and a target text, they tried to model whether it was a ‘same author’ or a ‘different author’ text pair, in the absence of any material for the author(s) of the source or target text, so that the algorithm could not even generalize over one author’s writing style across different documents. Thus, Koppel et al. formalized the task of authorship verification as a binary classification problems on the level of the text pair: given any two texts, an algorithm is asked to determine whether or not these texts were written by the same individual, a setup which is comparable to fingerprint authentication in biometrics. Interestingly, while this setup is perhaps extremely difficult, it is also universal, in the sense that *any* authorship problem can be reformulated in this way. Therefore, it is this approach which we will also use below.

**Preprocessing, vectorization and distance metrics**

Perhaps strikingly, most approaches in authorship attribution rely on relatively simple distance metrics, as opposed to e.g. probabilistic alternatives such as Naïve Bayes. These distance metrics serve to compare texts with respect to authorial style, and texts that are more similar in writing style, should yield a smaller distance according to a good distance metric. Burrows’s Delta, for instance, involves the calculation of a simple distance metric between an anonymous texts and all the writing samples available for the candidate authors: after comparing these distances, the anonymous text will be attributed to the author of sample to which the anonymous item which is closest. Argamon convincingly demonstrated how Delta (and many related measures) can therefore be considered a form of nearest-neighbour classification (also known as ‘kNN’, ‘memory-based learning’ or ‘lazy learning’), in which attribution is realized through the extrapolation of the class label of a test item’s nearest neighbor (i.e. *k*=1).

To be able to apply such distance metrics to (pairs of) documents, it is crucial to obtain a good numeric representation of the documents. The process of converting textual documents to such a representation is known as ‘vectorization’ and typically results in a two-dimensional matrix, in which each row represent a document and each column represent a particular feature (e.g. the relative frequencies of a particular word). Such a two-dimensional representation is called a ‘vector space model’, because a corpus is represented as a ‘space’ in which each individual document is ‘modeled’ by its own ‘vector’. In this paper, we will explore three commonly used vectorization strategies. The first model we will use is the simple ‘term-frequency model’ (*tf*), in which the model is identical to a simple frequency table, in which the cells are populated with the relative frequencies of features in texts. Secondly, we will use the ‘Term Frequency-Inverse Document Frequency model’ (*tfidf*): this model takes the simply *tf*-matrix as input, but scales the relative frequencies by dividing them through the inverse of an item’s document frequency in the entire corpus. This model has the interesting property that it will assign relatively more weight to rare features, which only occur in a limited number of documents. Thirdly, we will use the ‘Standard Deviation model’ (*std*), in which the original *tf*-matrix is scaled by the standard deviation of feature columns (Burrows, Argamon). As opposed to the *tfidf* model, the *std* model will increase the weight of features that have relatively stable frequencies in the corpus, such as common function words, and scale down the importance of rare items.

As to the preprocessing of the material, we have lowercased the diplomatic editions in the original documents and only considered alphabetical tokens. [Rik, kan je hier wat zeggen over de verwijdering van namen?] From the documents, we have extracted so-called ‘character ngrams’ or consecutive slices of characters from words. Word bigrams for instance are ngrams of size *n*=2, from the word ‘bigram’, for instance, we can extract the following ngrams: ‘ b’, ‘bi’, ‘ig’, ‘gr’, ‘ra’, ‘am’, ‘m ’. Note how word boundaries also get implicitly modeled in this representation, by allowing space characters to be included, although we did not allow ngrams to cross word boundaries. Character ngrams offer the succinct advantage that they can capture information beyond the simple lexical levels, and are thus also sensitive to morphemic and orthographic qualities of texts. In previous studies, character ngrams have often yielded state-of-the-art results in authorship studies.

In this paper, we will compare three commonly used distance metrics, which can all be freely combined with the vector space models described above: firstly, the Manhattan city block distance or the sum of the absolute difference of two vectors; secondly, the Euclidean distance which is equal to the length of the ordinary line between two vectors in the Euclidean space and thirdly, the Ruzicka minmax distance. The latter is a fairly recent addition to stylometry and previous research has convincingly demonstrated its remarkable qualities in authorship studies. When comparing two vectors, the metric will first collect the pairwise minima and maxima of both vectors. Next, it will compute the ratio of the summed minima and summed maxima and subtract the result from one (to obtain a distance metric, instead of a similarity metric). Note that the Euclidean metric is closely related to the cosine similarity between two vectors, which is also a commonly used metric in stylometry.

**Two verification methods**

In this paper, we will compare two recently introduced verification systems: the *pairwise* method and the *imposters* method. The first method is a fairly simple and perhaps naïve method, although it can be surprising difficult to beat. For the pairwise method, we proceed as follows. We exhaustively combine all documents and label them as ‘same author’ or ‘different author’ documents pairs (SADPs and DADPs). Next, we represent these documents using one of the vector space models (*tf*, *tfidf* or *std*) and we calculate the distance between each document pair (DP). Next, we assume that SADPs will generally yield lower distances according to these metrics and we determine a threshold which optimally separates the DADPs and SADPs. For testing purposes, we are confronted with the problem that corpora will typically contain much more DADPs than SADPs. Thus, a simple baseline systems which would label all DPs as ‘different author’ pairs, would be extremely hard to beat. For testing purposes we therefore follow the evaluation proposed by Koppel et al.: first we select all the SADPs which can harvested from the corpus, and then we randomly select an identical number of DADPs from the material. Throughout the paper, we will evaluate the fitness of our model (i.e. the performance of the optimal threshold) using the F1-score, which is an established evaluation measure in Machine Learning.

The *imposters* method is a slightly more complicated extension of the *pairwise* method, which resulted from the general worry that the pairwise method might suffer from overfitting on specific features in texts: if two short texts on the same topic coincidentally use the same low-frequency, highly topic-specific word, this might dramatically decrease the final distance measure, even if the texts are not authored by the same individual. Therefore, recent studies have suggested a bootstrapped approach to this problem, where the distance measure would calculated a number of times, but in each iteration, only a randomly selected subset (e.g. 50%) of the available features would be used (Koppel et al., Eder). The idea is that, for two texts to be attributed to the same person, the would have to be similar across a large number of randomly selected subsets of the feature space. Apart from this bootstrap-like feature sampling, the verification which we will use below additionally uses a background corpus of similar documents. The idea is that if two texts were written by the same author, they should be consistently more similar to each other than to other randomly selected texts.

Before running the feature sampling procedure, the verification procedure will therefore selected a subset of texts from a background corpus, namely the set of texts which are most similar to the source document according to the distance metric used, applied to the original set of features. These texts are called the *potential imposters*: during each iteration, we will now randomly select a smaller subset of *actual imposters* from this set of *potential imposters*. Given the source document, a distance metric and a random subset of the original features, we now calculate the distance between source document, on the one hand, and, on the other hand, the target document and the *actual imposters*. We then rank the target document amidst the *actual imposters*, and calculate its Mean Reciprocal Rank, or the multiplicative inverse of the target document’s rank. After this iterative procedure, we collect the average MRR which the target document obtained amidst the randomly selected imposters. This measure can be expected to reliably reflect how similar the source and target documents are, not only across various subsets of the entire feature space, but also when compared to various subsets of similar texts.

Instead of thresholding the system on the direct distance between the source and target document, we threshold the *imposters* system on the average MRR scores of the target document. While the imposters method can intuitively be expected to produce more stable results than the naïve pairwise method, previous research has demonstrated that the success of the verification procedure largely depends on the quality of the background corpus. Crucially, the background texts by imposter authors have to be similar enough to the actual corpus, in order to introduce enough confusion when comparing texts which were not authored by the same writer. On the other hand, if the background corpus is too similar to the test corpus, it might also introduce confusion when comparing texts which are in fact by the same author.

**Testing the systems**

The present epistolary corpus of soldier letters is characterized by an absolute lack of ground truth, in the sense that we little to no information on the (potentially collaborative) authorship of these documents. Nevertheless, to explore of the effect of different settings of our verification system, we have applied the following procedure. Many of the letters in the corpus have been signed by their senders: because we can assume that this signature will in many cases coincide with, at the very least, the *auctores intellectuals* of these documents, we have attempted to fit the system on these signatures as author labels, leaving out the anonymous documents where we have no information whatsoever on their authorship. We have therefore selected all signed letters from the corpus and uniformized the orthography of their signatures (e.g. … > …). To be able to test at least one SADP for each author, we have artificially create two documents from each letter: we randomly scrambled the word order in these letters and splitted them into two equal-sized documents and treated these as individual texts. From this dataset, we were able to extract 176 SADPS, which we complemented with 176 randomly select DADPs (352 DP in total). We counted 4,259 unique character tetragrams in these texts.

We start with testing the pairwise method, which is a fairly simple verification system, but nevertheless one which can be surprisingly hard to outperform. We have initially applied to the entire vocabulary of character tetragrams in this datasets (*n*=4; cf. Koppel et al.) to each combination of a vector space and distance metric, in order to test under which parametrization, the pairwise system could achieve an optimal fit. We report our results in Figure 1, which offers an intuitive visualization of the difference in distribution with the pairwises distances for SADPS (in blue, preferably small) and DADPs (in red, preferably large), using KDE density plots. In the Figure 1, we additionally we report the F1-score of the optimal threshold, as well as the outcome of a Kolgomorov-Smirnov test, testing the size of the difference between the SADPs and DADPs. Overall, it becomes clear that Manhattan distance yields the worst results and that the *std* vector space model is the best vector space. While the Euclidean distance clearly rivals the results of the minmax measure, the latter proves to be more stable across vector space. These results are in line with previous research.

Additionally, we have tested the effect of applying the combination of the minmax model and the *std* vector space to different (truncated) vocabularies, spanning various sets of the most frequent character ngrams in the corpus. For various sizes of *n* (2-5), we report F1-scores for different vocabulary sizes, spanning 100 equally-sized intervals of features between 30 and 4,259. As shown in Figure 2, adding more features generally helps and the combination of *std*+minmax can deal surprisingly well with the increasingly sparse vectors. Except for bigrams (which of course yield a much smaller vocabulary), the ngram sizes from 3-5 yield similar results, although the results for tetragrams appear to be slightly higher.

Note that the F1-scores of the pairwise method are already quite high (in the lower .90s), so that the impostor method will probably have a hard time improving upon these results. In Figure 2, we show a density plot which is similar to the one in Fig. 1, although the distribution here does not reflect a direct pairwise distance, but rather the average MRR. Again, we used the entire vocabulary of character tetragrams, using 50 initial *potential* imposters and 10 *actual* imposters (i.e. in each iteration). With respect to the feature bootstrapping, we selected 50% of the original feature set. Importantly, we did not allow any texts signed by one of the authors of the target or source document into the imposter set. [Hieronder geef ik de resultaten voor 3 achtergrondcorpora, zijnde: het testcorpus zelf (dat het momenteel het beste doet, met een minimale verbetering in het geval van minmax+tfidf), de armenbrieven, en de *sailing letters*. We moeten waarschijnlijk eentje kiezen. Wat me opvalt is dat de armenbrieven het eigenlijk relatief goed doen als impostors, hoewel het er eigenlijk niet veel zijn: mochten er hier meer van zijn te verzamelen (Rik?), ziouden resultaten waarschijnlijk nog hoger liggen.]

Macintosh HD:Users:mike:GitRepos:verification:soldier_experiments:outputs:distribs_baseline.pdf

Figure 1

Macintosh HD:Users:mike:GitRepos:verification:soldier_experiments:outputs:featrngs_baseline.pdf

Figure 2

Achtergrondcorpus = testcorpus

Macintosh HD:Users:mike:GitRepos:verification:soldier_experiments:outputs:distr_impost(soldier_letters_as_background).pdf

Achtergrondcorpus = armenbrieven

Macintosh HD:Users:mike:GitRepos:verification:soldier_experiments:outputs:distr_impost(soldier_armen_as_background).pdf

Achtergrondcorpus = sailing letters

Macintosh HD:Users:mike:GitRepos:verification:soldier_experiments:outputs:distr_impost(soldier_sailing_as_background).pdf