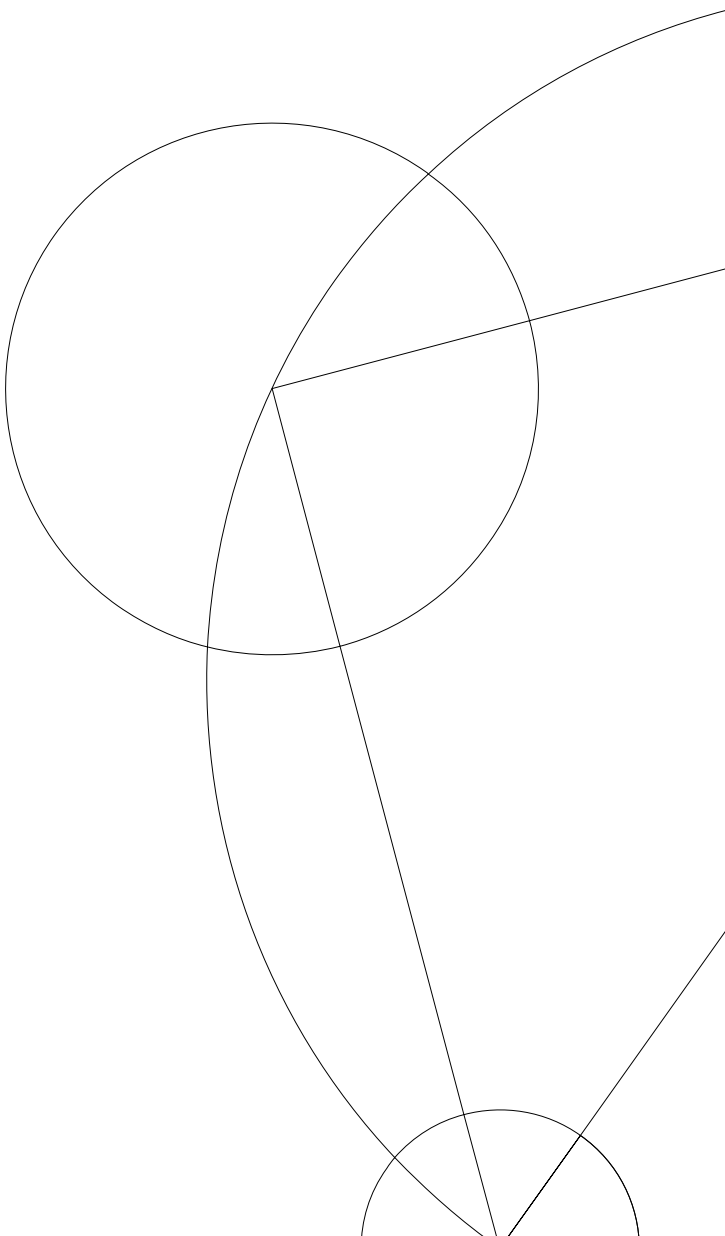
UNIVERSITY OF COPENHAGEN

# Authorship Verification

Project Outside Course Scope

August V. Sørensen & Magnus N. Stavngaard
august.vinkel@gmail.com magnus@stavngaard.dk
NCB360 MZC887

February 28, 2018



**Abstract**

In this report we investigate different methods for authorship verification. Authorship verification is the process of determining whether a text is written by an author given a set of texts written by the same author. We will implement a select few of the algorithms we investigate. The specific algorithms are:

First, the Delta Method which we use as a baseline for the other methods we implement. The Delta Method is a distance based approach that use features (vector of numbers) extracted from the texts and a distance metric to find the closest author of an unknown text.

Second, a generalising Random Forest approach is implemented. The method encodes features extracted from the texts using a Universal Background Model (UBM) and applies the Random Forest to those encoded features.

Third, the Delta Method is expanded by trying different features and distance metrics than used in the original Delta Method.

Fourth, an Author Specific Support Vector Machine (SVM) is implemented. The SVM is trained using the imposter method. Two sets of texts are generated. One set of texts known to be written by the author and another set of texts known to not be written by the author. Features are then extracted from all texts and a SVM is then trained on those sets of features.

For training and evaluation we use two datasets. The datasets are from two instances of a yearly competition in text forensics (PAN). Specifically we use the dataset from the 2013 edition and the 2015 edition. We obtain the third best result on the PAN 2013 task and the eights best result on the PAN 2015 task.

# Contents

**1 Introduction** **1**

**2 Related Work** **1**

**3 Method** **4**
3.1 Text Features . . . . . 5
3.2 Delta Method . . . . . 5
3.3 Generalising Random Forest . . . . . 5
3.4 Extended Delta . . . . . 6
3.5 Author Specific SVM . . . . . 6
3.6 Experiments . . . . . 7
3.6.1 Delta Method . . . . . 7
3.6.2 Generalising Random Forest . . . . . 10
3.6.3 Extended Delta . . . . . 10
3.6.4 Author Specific SVM . . . . . 13

**4 Results** **13**
4.1 Delta Method . . . . . 14
4.2 Generalising Random Forest . . . . . 14
4.3 Extended Delta . . . . . 15
4.4 Author Specific SVM . . . . . 17

**5 Discussion** **17**
5.1 Comparison to baseline approach . . . . . 17
5.2 Comparison to other PAN results . . . . . 17
5.3 Feature Importance . . . . . 19
5.4 Performance of our different methods . . . . . 21
5.5 Improvements . . . . . 23

**6 Conclusion** **24**

**7 Future Work** **24**

**A Pan 2013 Results** **27**

**B Pan 2015 Results** **27**

# 1 Introduction

Authorship verification is the process of verifying the authorship of a text. You are given a set of texts known to be written by the author and an unknown text that has to be classified as either written by the author or someone else. The normal approach to the problem is to use stylometry to extract features from the text and then some form of machine learning or statistical method to analyse the data [Stamatatos, 2009]. Many different text features have been proposed to describe an author's writing style. That includes, but is not limited to, character frequencies, word frequencies, vocabulary size, sentence length, punctuation usage, character n-grams, word n-grams and Parts-of-Speech (POS)-tagging n-grams [Stamatatos, 2009].

Authorship verification has been used for plagiarism control in danish secondary schools [Hansen et al., 2014] and also has uses in civil law, criminal law and computer forensics [Stamatatos, 2009].

In this article we explore state-of-the-art methods for authorship verification. We extract several types of features including word frequencies, word n-grams and POS-tagging n-grams. We implement several algorithms to work on the features. We start by implementing a baseline method which is a distance based approach in which an unknown text is considered to be written by the closest author. We then use the baseline method to compare to our other results. The other results are obtained from several different methods. A Random Forest based method, an extension of the baseline method and a SVM based method. We use data from two sources [Stamatatos et al., 2015] and [Efstathios Stamatatos et al., 2013] which are two instances of a yearly competition in digital text forensics. Since we use the data from those two competitions we also compare our results to the results obtained by others in the competition.

All code produced for this report can be found in the following git repository:
`https://github.com/Smazle/Authorship-Verification`

# 2 Related Work

[Stamatatos, 2009] gives a really good overview of the current state and history of authorship verification and authorship attribution methods.

PAN [1] keeps a collection of shared tasks in digital text forensics. In 2013, 2014 and 2015 the tasks focused on authorship verification among other things. In this report we work with data from PAN 2013 and PAN 2015. In the 2013 task a dataset of authors were given. Each author had a collection of known texts and a single unknown text. The task was to determine which of the unknown texts was written by the same author as the known texts it was grouped with. In the 2015 task a dataset of authors was also given. However, each author had only a single known text and a single unknown text, and the task was to determine which of the unknown texts belonged to the same author as the known texts. The PAN 2013 task was ranked by the F1 measure and the PAN 2015 task was ranked according to the Area Under Receiver Operating Characteristic (AUROC) and the c@1 measure [Peñas and Rodrigo, 2011]. We describe the measures in more detail in Section 4. In this section we describe many approaches used to solve the PAN 2015 task. When we report a "final score" we mean the product of the two measures for the PAN 2015 task as that was the final measure they were ranked by. The results of the PAN 2013, and PAN 2015 competitions on the English texts are found Appendix A, and B respectively.

[1] `http://pan.webis.de/`

[Posadas-Durán et al., 2015] chose to perform their feature extraction on the syntactic level. This was done using syntactic n-grams, which was extracted using a syntactic analyser designed for the designated language. After extracting the n-grams wanted, filtering was performed, removing the less frequent n-grams. At this point they chose to represent their n-gram frequencies as a vector, allowing them to use the Jaccard distance to measure the difference between new introduced unknown texts, and their proposed authors' known texts. When the similarity fell under a certain threshold, the author was deemed non-valid. This yielded a final score of 0.39999 on the English texts, and [Posadas-Durán et al., 2015] noted that a new heuristic handling ill-constructed sentences would probably have improved their results, as they were just discarded in their case. Additionally, more features describing other linguistic layers, such as lexical, and syntactic features would probably improve their results as well.

[Maitra et al., 2015] implemented a solution for the PAN 2015 task. They used a collection of different features extracted from the known texts. The features were based on punctuation, sentence length, vocabulary, character n-grams and POS tagging. They trained a Random Forest Classifier on the features extracted and used that to determine whether or not the unknown texts were written by the author. The final score of the method was 0.34749 on English texts, which is not overwhelming and they commented that deep learning might make their results better.

[Pacheco et al., 2015] also proposed using a random forest for the PAN 2015 task. They implemented two baseline models and one real model. The baseline models were a simple distance metric with a trained cutoff point and a Gaussian Mixture Model. The second baseline model was about defining a general feature vector for all authors and a feature vector for each specific author. To determine if a text from an unknown author is written by a given author, you compute the distance between the texts feature set and the universal and author specific feature set. If the unknown text is closer to the universal text than to the author specific text it is presumed to not be written by the author. The used features in this case were number of stop words, number of sentences, number of paragraphs, spacing, punctuation, word frequencies, character frequencies, punctuation frequencies, lexical density, word diversity, unique words and unique words over all authors. The main model made use of a random forest and a UBM. A feature vector was again computed for all the known texts in the dataset and used to construct the UBM. I addition to that, a feature vector was computed on each individual known text of the dataset. Each of these author specific feature vectors were then encoded using the UBM. The encoded vectors were then combined, with a known and unknown text after which the result were fed to a Random Forest model. Their final score was 0.43811 on English texts.

[Bartoli et al., 2015] proposed yet another Random Forest based approach. They didn't use a Random Forest Classifier as [Maitra et al., 2015] and [Pacheco et al., 2015] but a random forest regressor. The used features were word n-grams, character n-grams, POS tag n-grams, word lengths, sentence lengths, sentence length n-grams, word richness [2], punctuation n-grams and text shape n-grams. They then performed a feature selection and normalization. They performed the final regression with both trees, a Random Forest and a SVM. They ended up choosing the Random Forest as it performed the best. Their results were very good, having the best final score on Spanish texts. However their English final score was only 0.323.

[Castro et al., 2015a] present an approach that focuses more on the feature extraction, than the algorithm applied to it. They made use of a set of 10 features spanning across 3 different linguistic layers, the character layer, the lexical layer and the semantic layer. For each of these 10 features a vote is cast. The vote is determined by comparing the average similarity of the authors' texts. When classifying an unknown text as either written by

[2]Word richness is number of distinct words in a text divided by the total number of words.

the same author or another author the following steps are performed: For each author, the similarity of their texts are computed using one of the 10 feature vectors as input for a chosen similarity function. The similarity for each of these authors are then averaged to form the Average Group Similarity (AGS). The new document is then added to the group of documents of the proposed author, and the similarity of that new group of documents is computed. If that similarity is above the AGS, the unknown text will be classified as being written by the author, and a vote for this decision is thrown. However, before the vote for the feature is finalized, it is done with 3 different similarity functions, Cosine, Dice and 1-MinMax. The majority vote of these 3, similarity functions, determine what the vote should be on that specific feature. This is done for each of the 10 features, where in case of a tie vote (5 against 5) no decision is taken. This yielded some very good results, however some questions were raised as to the accuracy of documents on other genres. The final score of this average based voting method, ended up being the second best in the PAN 2015 competition with a score of 0.52041 on English texts.

[Gutierrez et al., 2015] use a somewhat different approach by using Homotopy-based Classification in their work. Their approach use a 4 different features. Bag of words, bigram of words, Punctuation and trigram of words. From there a set of imposters are created, using the generic imposter method. The $L$-1 homotopy is then applied, constructing a feature set. That feature set will be matching a document generated using the imposters and a known text. The unknown document and the reconstructed document are then compared using what is called the computed residual. This residual is compared to each author in the set, and if it doesn't match the proposed author, the author isn't considered the writer of the unknown text. The method performed well for all the languages used in their test besides Dutch which they explain is because of the short texts provided. The final score was 0.51.

[Gómez-Adorno et al., 2015] solved the PAN 2015 task by using a graph based approach. The graph used is an Integrated Syntactic Graph (ISG) which represents the text by creating a graph for each sentence and combining those graphs into one large graph. The authors constructed such a graph for each text and used commonalities in shortest paths in the graph to compare the texts. The results were not very good, relative to the other entries in the competition, with a final score of 0.2809.

[Layton, 2014] makes use of a more simple approach. He makes use of different types of N-Grams. These N-Grams was used to compute a feature vector for each document written by a specific author. Collectively they can be combined to a matrix describing the writing style of that author. When a new unknown text is introduced, one of three comparative algorithms is used to compute the average similarity between each of the authors' known texts (Intra-Distance). The average similarity between the unknown text and the known ones by the proposed author is then computed as well (Inter-Distance). The author is considered correct if the Inter-Distance is lower than the Intra-Distance plus 2 times the known datasets' standard deviation. This was performed using different comparative metrics. It did however, not perform very well having a final score of 0.36277.

[Castro et al., 2015b] solved the PAN 2014 task by using the average similarity of an unknown text to known texts of an author. The features used were character n-grams, character n gram prefixes, character n gram postfixes, word n grams, punctuation, POS tagging n grams, POS tagging at start of sentences and POS tagging at the end of sentences. [Castro et al., 2015b] tried several different similarity measures Cosine, Dice, Jaccard, Tanimoto, Euclidean and MinMax. They generally got the best results with Dice and Jaccard similarity. Their approach was only applied to spanish texts in this case, and as such the results aren't relevant for comparison in our case.

[Hansen et al., 2014] and [Aalykke, 2016] both describe usage of authorship attribution methods in identifying authors of texts written in Danish secondary schools. [Aalykke, 2016] mainly used a distance based approach. They extracted features and then used different

distance metrics to compute the closest and therefore best author. [Hansen et al., 2014] used SVMs for the author classification. They obtained an accuracy of 84% .

# 3 Method

In this section we will describe the methods we have implemented to solve the PAN 2013 and PAN 2015 problems. Both problems are known as authorship verification problems. [Stamatatos, 2009] describes the problem in a machine learning sense as a binary classification problem. Either the unknown texts are written by the same author or they are written by different authors. All our implemented methods solve that classification problem by answering either *True* (unknown text is written by the same author as the known texts) or *False* (unknown text is **not** written by the same author as the known texts). Since both PAN 2013 and PAN 2015 are a binary decision problem we can compute the number of True Positive (TP)s, True Negative (TN)s, False Positive (FP)s and False Negative (FN)s. In these problems we get,

- a TP whenever we answer *True* and the texts are written by the same author,
- a TN whenever we answer *False* and the texts are **not** written by the same author,
- a FP whenever we answer *True* and the texts are **not** written by the same author,
- a FN whenever we answer *False* and the texts are written by the same author.

Given those definitions the True Positive Rate (TPR), False Positive Rate (FPR), True Negative Rate (TNR) and False Negative Rate (FNR) describes.

**TPR:** The fraction of positives that we reported *True* on i.e. the fraction of texts written by the same author that we say are written by the same author.

**FPR:** The fraction of negatives that we reported *True* on i.e. the fraction of texts written by different authors that we say are written by the same author.

**TNR:** The fraction of negatives that we reported *False* on i.e. the fraction of texts written by different authors that we say are written by different authors.

**FNR:** The fraction of positives that we reported *False* on i.e. the fraction of texts written by the same author that we say are written by different authors.

And they can be computed as,

$$TPR = \frac{TP}{TP + FN}, \tag{1}$$

$$FPR = \frac{FP}{FP + TN}, \tag{2}$$

$$TNR = \frac{TN}{TN + FP}, \tag{3}$$

$$FNR = \frac{FN}{FN + TP}. \tag{4}$$

[Stamatatos, 2009] describes instance based and profile based approaches for authorship attribution/verification. The instance based approach uses several texts from an author to create multiple different feature vectors representing the authors' writing style while the profile based approach concatenates all texts to a single text and creates only a single vector

representing that text. The instance based approach corresponds to the PAN 2013 problem. In that problem we are given multiple texts to train from and the methods we have implemented to solve that problem will reflect that property. The profile based approach corresponds to the PAN 2015 dataset; here we are given only a single known text per author, which we can use to construct a profile of the author.

### 3.1 Text Features

Feature extraction from a text is the process of finding a vector that represents that text. Many different features can be extracted when performing Natural Language Processing (NLP). They can span many different linguistic layers, including but not limited to, layers such as the character layer, which describes a text on the character level, and the phonetic layer which describes a text based on the phonetic alphabet. We use features spanning several of these layers. Specifically we use some character level features, some word level features and some POS tagging features. In this section we will define the different features we use. The specific features used in different experiments will be described at length under those experiments.

N-grams are subsequences extracted from a sequence of tokens. For example, 3-grams are all subsequences of length three of a given sequence. Using individual characters as tokens, all *character 3-grams* of the string "hello" are "hel", "ell" and "llo". We use several different types of n-grams in different experiments including character n-grams, word n-grams, special character n-grams and POS-tagging n-grams. Word n-grams are subsequences of words, special character n-grams are subsequences of characters with alphanumeric and space characters removed and POS tagging n-grams are subsequences of POS tags which are word classes such as nouns, verbs and adjectives. A special case of n-grams is 1-grams which is just a count of the different tokens in a sequence. We will refer to 1-grams as frequencies.

### 3.2 Delta Method

We have chosen to use the Delta method as a baseline for our other implementations. The Delta method is described by [Evert et al., 2015] and consists of first extracting word frequencies, then applying a linear transformation to those frequencies and finally using K-Nearest Neighbours (KNN) with different distance metrics. There are a number of parameters to choose. The main parameter, is choosing how many word we are going to consider when finding the word frequencies. The amount of words was originally chosen at 150 words [Evert et al., 2015]. Then the linear transformation can be chosen, the usual transformation is a normalization to zero mean unit variance. And finally the distance metric can be chosen. The distance metric used in the original Delta Method was the Manhattan distance [Evert et al., 2015].

### 3.3 Generalising Random Forest

In addition to the Delta Method we chose to use the Random Forest approach suggested by [Pacheco et al., 2015]. In our implementation we use different features than in the their proposal. Their idea was to make a more generalized approach, which would base itself on the comparison of a general model rather than an author-specific one. The point was to get around the data constraints of the PAN 2015 dataset. This was to be be done, by combining the unknown text, the known text, and then a third general model of all known texts, using a specific encoding. Rewriting the encoding function used in [Pacheco et al., 2015], we get the following combining function,

$$R_{i_k} = \frac{(A_{i_k} - U_{i_k})^2 + 1}{(B_k - U_{i_k})^2 + 1} \tag{5}$$

Where $A$ is a known text, $U$ is an unknown text, $B$ is a UBM describing a general author independent text. $i$ denotes from which dataset-entry we get our known and unknown texts, and $k$ denotes a specific feature of a given text. As such $R_{i_k}$ describes the encoding of a specific feature for a specific data point.

The UBM is meant to represent the features of an author *independent* text. It is computed by concatenating all known texts in the training dataset and computing features from that resulting text. Since multiple authors are then part of the text we are computing features from, the assumption is that the author specific features will be averaged out and the UBM will represent the features of an author independent text. The addition of 1 in the model prevents division by zero. The squaring of $(A_i - U_i)$ and $(B - U_i)$ both prevents negative values and punishes values that are far away. Therefore each value in the resulting encoded feature vector is in the range $[0; \infty[$.

Let's examine what the Equation (5) describes. Fix any specific $i$ and let $A$ be $A_i$ and $U$ be $U_i$ then for each feature $k$ we compute,

$$R_k = \frac{(A_k - U_k)^2 + 1}{(B_k - U_k)^2 + 1}, \tag{6}$$

The $k$'th feature in each of the vectors is the same feature just extracted from different texts. When the feature of the unknown text $U_k$ is closer to $A_k$ than $B_k$ the numerator in the fraction will be greater than the denominator giving us something in the range $[0; 1]$. When the feature of the unknown text $U_k$ is closer to $B_k$ than $A_k$ the numerator will be lesser than the denominator and we will therefore get a value in the interval $[1; \infty[$. That results in $R$ being a vector containing values from 0 to $\infty$ where it is between 0 and 1 whenever a feature is closer to the author specific text than the universal text and greater than 1 otherwise.

The random forest algorithm is then trained on these encoded feature vectors where it is supposed to learn a general difference between feature vectors of the same author and feature vectors of different authors.

### 3.4 Extended Delta

As described earlier there are many ways to extend and change the delta method. We tried both using different features and different distance measures to obtain better results than the Delta Method described in Section 3.2. The different feature combinations were tried experimentally one after the other. If a feature combination did well on a dataset we tried adding or removing features from that feature set to maybe obtain something better. The features we tried were different combinations of character n-grams, word n-grams and POS-tag n-grams.

The different distance measures we tried were the Manhattan distance and the Euclidean distance. We chose the Manhattan distance since it has been shown to consistently perform well when compared to other distance metrics [Evert et al., 2015]. And we chose the Euclidean metric since it is very easy to implement and it performs about as well as the Manhattan distance, on a small amount of features [Evert et al., 2015].

### 3.5 Author Specific SVM

We implemented an approach using a SVM inspired by [Hansen et al., 2014]. The approach is only applicable to problems with more than a single text per author. The classification in this approach is done by training an SVM classifier on all known texts of an author and an equal number of texts from other authors. Then the unknown text is given to the SVM and is classified either as belonging to the same author or as belonging to a different author.

If there is only a single known text available for an author it does not make sense to train an SVM since there is simply too little data for it to be a viable choice.

### 3.6 Experiments

In this section we describe the different experiments we have performed. We have tested the different methods we have implemented with different features and on different datasets. In our experiments we use data from PAN 2013 [3] and PAN 2015 [4].

The PAN 2013 data consists of texts from English, Greek and Spanish authors. We work only on the English texts which, of which there are 10 authors. For each author there is (between 1 and 10) known texts and a single unknown text. The task is, given the known texts of an author, to determine whether the unknown text is written by the same author.

The PAN 2015 data consists of texts by authors in English, Dutch, Greek and Spanish. Again we only work with English texts. Unlike the 2013 dataset there is only a single known text for each author and a single unknown text. There is therefore much less known data available for each author which makes the verification harder. In the 2015 dataset there are 100 *problems* in which some of the known texts are the same (i.e. there are not 100 authors). The PAN 2013, and PAN 2015 text, have an average word count of 1038, and 460 respectively.

To generate metadata about English texts we use the Brown dataset [5]. The Brown dataset contains more than 1,000,000, words organized in sentences, across different genres and is therefore perfect to use for our datasets. We specifically use the dataset to identify the most frequently used n-grams (of all kinds). If we were to use our training data to generate that, we would risk having a bias towards our specific training dataset.

We will evaluate the performance of the different algorithms on the training data with the accuracy of the algorithms. The accuracy is computed as the number of correct answers divided by the total number of problems. For some methods we will also report the TPR and TNR.

#### 3.6.1 Delta Method

In the Delta Method we work only with word frequencies as originally proposed. As the linear function we normalize to 0 mean and unit variance and as features we use the word frequencies of the $n$ most frequent words. We get the most frequent words by using the Brown dataset. In the KNN part we use the Manhatten distance and only a single nearest neighbour since in one of the datasets we have only 1 text for each author. To classify the unknown texts as either written by or not written by the author we train a KNN for each unknown text. Each classifier is trained with a known text from the author in question and $m$ other random texts. If the unknown text is classified as belonging to one of the $m$ random authors instead of the author in question we report that the unknown text is not written by the author. If the text is classified as belonging to the author in question we classify it as being written by the author.

We chose the number of most frequent words $n$ and number of opposing authors $m$ by trying different configurations in a grid and choosing the best values. Each configuration is tried 100 times since random authors are chosen in each run and averaged. On the training dataset we obtained the results shown in Table 1 for the PAN 2013 data and in Table 2 for the PAN 2015 data. For PAN 2013 the results are generally better since there is more text available and the best accuracy was obtained when using the 300 most frequent words and 4 opposing authors. For PAN 2015 the best result is obtained when using the 200 most frequent words and 1 opposing author.

[3] http://pan.webis.de/clef13/pan13-web/index.html
[4] http://pan.webis.de/clef15/pan15-web/index.html
[5] http://clu.uni.no/icame/brown/bcm-los.html

**PAN 2013 Delta Method Results**

| | | $n = 100$ | $n = 200$ | $n = 300$ | $n = 400$ | $n = 500$ |
|---|---|---|---|---|---|---|
| $m = 1$ | **Accuracy** | 0.62093 | 0.64437 | 0.65062 | 0.66718 | 0.66281 |
| | **TPR** | 0.75812 | 0.79562 | **0.80812** | 0.79750 | 0.77500 |
| | **TNR** | 0.48375 | 0.49312 | 0.49312 | 0.53687 | 0.55062 |
| $m = 2$ | **Accuracy** | 0.65375 | 0.68562 | 0.69593 | 0.68968 | 0.68250 |
| | **TPR** | 0.63250 | 0.71875 | 0.74000 | 0.70062 | 0.67125 |
| | **TNR** | 0.67500 | 0.65250 | 0.65187 | 0.67875 | 0.69375 |
| $m = 3$ | **Accuracy** | 0.66250 | 0.68187 | 0.69687 | 0.69250 | 0.69843 |
| | **TPR** | 0.55937 | 0.64125 | 0.68375 | 0.65437 | 0.63375 |
| | **TNR** | 0.76562 | 0.72250 | 0.71000 | 0.73062 | 0.76312 |
| $m = 4$ | **Accuracy** | 0.66312 | 0.68656 | **0.70062** | 0.68062 | 0.68281 |
| | **TPR** | 0.49875 | 0.61125 | 0.65125 | 0.61875 | 0.57562 |
| | **TNR** | 0.82750 | 0.76187 | 0.75000 | 0.74250 | 0.79000 |
| $m = 5$ | **Accuracy** | 0.66343 | 0.69468 | 0.69250 | 0.66937 | 0.68437 |
| | **TPR** | 0.46687 | 0.59375 | 0.61812 | 0.57500 | 0.54937 |
| | **TNR** | 0.86000 | 0.79562 | 0.76687 | 0.76375 | 0.81937 |
| $m = 6$ | **Accuracy** | 0.63531 | 0.67437 | 0.69406 | 0.67312 | 0.66718 |
| | **TPR** | 0.39937 | 0.55125 | 0.59750 | 0.56750 | 0.50312 |
| | **TNR** | 0.87125 | 0.79750 | 0.79062 | 0.77875 | 0.83125 |
| $m = 7$ | **Accuracy** | 0.63843 | 0.67906 | 0.68437 | 0.67000 | 0.66562 |
| | **TPR** | 0.37875 | 0.53812 | 0.58500 | 0.54687 | 0.49625 |
| | **TNR** | 0.89812 | 0.82000 | 0.78375 | 0.79312 | 0.83500 |
| $m = 8$ | **Accuracy** | 0.63406 | 0.69281 | 0.68156 | 0.65218 | 0.67156 |
| | **TPR** | 0.36937 | 0.55062 | 0.56250 | 0.52250 | 0.50000 |
| | **TNR** | 0.89875 | 0.83500 | 0.80062 | 0.78187 | 0.84312 |
| $m = 9$ | **Accuracy** | 0.63781 | 0.68031 | 0.66437 | 0.66656 | 0.67031 |
| | **TPR** | 0.36625 | 0.51937 | 0.53000 | 0.52562 | 0.47375 |
| | **TNR** | **0.90937** | 0.84125 | 0.79875 | 0.80750 | 0.86687 |
| $m = 10$ | **Accuracy** | 0.62562 | 0.68781 | 0.68343 | 0.67687 | 0.66000 |
| | **TPR** | 0.34625 | 0.51375 | 0.54937 | 0.53125 | 0.46312 |
| | **TNR** | 0.90500 | 0.86187 | 0.81750 | 0.82250 | 0.85687 |

Table 1: Accuracy, TPR and TNR on different amounts of most frequent words $n$ and different numbers of opposing authors $m$ for the Delta Method. Each number is an average of 100 runs since there is randomness involved when picking the opposing authors. The test is run on the PAN 2013 training dataset. Maximum values for both Accuracy, TPR and TNR is shown in bold.

**PAN 2015 Delta Method Results**

| | | $n = 100$ | $n = 200$ | $n = 300$ | $n = 400$ | $n = 500$ |
|---|---|---|---|---|---|---|
| $m = 1$ | **Accuracy** | 0.56950 | **0.60239** | 0.56210 | 0.57269 | 0.56170 |
| | **TPR** | 0.58740 | **0.64660** | 0.61759 | 0.62200 | 0.61020 |
| | **TNR** | 0.55160 | 0.55820 | 0.50659 | 0.52340 | 0.51320 |
| $m = 2$ | **Accuracy** | 0.56810 | 0.58970 | 0.56470 | 0.57290 | 0.56030 |
| | **TPR** | 0.42219 | 0.46740 | 0.45720 | 0.45340 | 0.44640 |
| | **TNR** | 0.71400 | 0.71200 | 0.67220 | 0.69239 | 0.67420 |
| $m = 3$ | **Accuracy** | 0.55399 | 0.57510 | 0.56220 | 0.56850 | 0.55380 |
| | **TPR** | 0.32480 | 0.36880 | 0.36060 | 0.36660 | 0.33820 |
| | **TNR** | 0.78319 | 0.78140 | 0.76379 | 0.77040 | 0.76940 |
| $m = 4$ | **Accuracy** | 0.54589 | 0.56390 | 0.54560 | 0.55580 | 0.54710 |
| | **TPR** | 0.25920 | 0.29940 | 0.29460 | 0.29000 | 0.27800 |
| | **TNR** | 0.83260 | 0.82840 | 0.79660 | 0.82160 | 0.81620 |
| $m = 5$ | **Accuracy** | 0.53520 | 0.55430 | 0.54150 | 0.56290 | 0.54960 |
| | **TPR** | 0.21000 | 0.24920 | 0.24420 | 0.25860 | 0.24220 |
| | **TNR** | 0.86040 | 0.85939 | 0.83880 | 0.86720 | 0.85700 |
| $m = 6$ | **Accuracy** | 0.52749 | 0.54820 | 0.53709 | 0.55530 | 0.54260 |
| | **TPR** | 0.17980 | 0.21840 | 0.21800 | 0.23100 | 0.20739 |
| | **TNR** | 0.87520 | 0.87800 | 0.85620 | 0.87959 | 0.87780 |
| $m = 7$ | **Accuracy** | 0.52790 | 0.54189 | 0.53240 | 0.54550 | 0.54530 |
| | **TPR** | 0.16240 | 0.19160 | 0.19440 | 0.18660 | 0.19020 |
| | **TNR** | 0.89340 | 0.89220 | 0.87040 | 0.90440 | 0.90040 |
| $m = 8$ | **Accuracy** | 0.52340 | 0.53570 | 0.52459 | 0.54899 | 0.53419 |
| | **TPR** | 0.14820 | 0.17440 | 0.16180 | 0.17760 | 0.15560 |
| | **TNR** | 0.89860 | 0.89700 | 0.88740 | 0.92040 | 0.91280 |
| $m = 9$ | **Accuracy** | 0.51930 | 0.53340 | 0.52060 | 0.54229 | 0.53460 |
| | **TPR** | 0.13040 | 0.15220 | 0.15200 | 0.15700 | 0.15060 |
| | **TNR** | 0.90819 | 0.91460 | 0.88920 | 0.92760 | 0.91860 |
| $m = 10$ | **Accuracy** | 0.51780 | 0.52550 | 0.52260 | 0.54080 | 0.53150 |
| | **TPR** | 0.11860 | 0.13760 | 0.13960 | 0.13660 | 0.12940 |
| | **TNR** | 0.91700 | 0.91340 | 0.90560 | **0.94500** | 0.93360 |

Table 2: Accuracy, TPR and TNR on different amount of most frequent words $n$ and different numbers of opposing authors $m$ for the Delta Method. Each number is an average of 100 runs since there is randomness involved when picking the opposing authors. The test is run on the PAN 2015 training dataset. Maximum values for both Accuracy, TPR and TNR is shown in bold.

### 3.6.2 Generalising Random Forest

When applying the Generalising Random Forest algorithm, we have the possibility to use many features, since the Random Forest algorithm selects the best features from a given set of features. Therefore we could provide it with a large set of features, and let the algorithm filter away the bad ones. This however comes at the cost of runtime. Thus the task of finding a good/perfect input to train our forest on is quite extensive.

The `sklearn` library [6] offers a wide variety of configurations with regards to how our forest is built. Most configuration was set to their default value, with the exception of `n_estimators`. `n_estimators` denotes how many decision trees are in our forest, and due to the ensemble nature of the algorithm, we can increase this parameter to create an average classification prediction based on a larger set of individual predictions, thus decreasing the overall variance our model. In the following experiments, `n_estimators` is set to 1000. The payoff when increasing the number of trees used is the runtime. The increase in trees also have diminishing returns on the decrease of variance, so at some point increasing the trees don't make any sense, as the variance is only altered very little with each new tree. 1000 trees were the number we deemed appropriate to adequately decrease the variance of the forest, while still running at an acceptable speed.

The dataset in these experiments was split into a training and a validation set. The selection was done by randomly shuffling the entire data set and taking the first 80% of the dataset as the training set, and the last 20% as the validation set. As that introduces randomness we run the algorithm 100 times and take the average.

In the following, we created our UBM using the concatenation of all author-specific texts, from our training dataset.

In terms of features, we chose to feed our random forest algorithm a large set of features with different focuses. The algorithm is going to pick features, based on their impact on the actual classification. The low-impact features aren't adding any noise, so we might as well feed the algorithm as many features as possible, and then let it make the decision based on their individual impact. We chose to use the following features:

- The 50 most frequent word-n-grams for $n \in \{1, \ldots, 5\}$.
- The 50 most frequent character-n-grams for $n \in \{2, \ldots, 5\}$.
- The 50 most frequent POS-tag-n-grams for $n \in \{2, \ldots, 5\}$.
- The 5 most frequent special-character-n-grams for $n \in \{2, 3\}$.

In order to achieve better generalization, the Brown corpus was used as the basis for the feature generation.

In addition to the UBM based encoding we also tried another encoding. The method is exactly the same as described above except Equation (6) was replaced with

$$R_k = A_k - U_k. \tag{7}$$

That yielded an accuracy of 0.5675.

### 3.6.3 Extended Delta

We tried different feature combinations and distances. The results of running on the training data is shown in Table 3 for the Manhattan distance and in Table 4 for the Euclidean distance.

[6] http://scikit-learn.org/stable/modules/generated/sklearn.ensemble.Rand omForestClassifier.html

**Manhattan Distance Extended Delta Method**

| Character n-grams | Word n-grams | POS-tag n-grams | Special character n-grams | PAN 2015 result | PAN 2013 result |
|---|---|---|---|---|---|
| 100 2-, 3-, 4-, 5-grams | 5 1-, 2-, 3-, 4-grams | 10 1-, 2-, 3-, 4-grams | 5 1-, 2-, 3-grams | 0.54680, 3 | 0.68031, 3 |
| 100 2-, 3-, 4-grams | 300 1-grams | NONE | NONE | 0.56340, 2 | **0.73718, 5** |
| 100 2-, 3-, 4-grams | NONE | NONE | 20 2-, 3-, 4-grams | 0.56980, 2 | 0.68500, 3 |
| 100 2-, 3-, 4-grams | NONE | 15 1-, 2-, 3-grams | 20 2-, 3-, 4-grams | 0.58029, 3 | 0.67718, 8 |
| NONE | NONE | 15 1-, 2-, 3-grams | NONE | 0.51470, 10 | 0.62062, 8 |
| 20 2-, 3-, 4-, 5-, 6-grams | NONE | NONE | NONE | 0.54640, 4 | 0.69000, 8 |
| 150 2-, 3-grams | NONE | NONE | NONE | 0.56370, 4 | **0.70531, 4** |
| NONE | 50 2-, 3-grams | NONE | NONE | 0.52120, 9 | 0.60812, 3 |
| NONE | 20 2-, 3-grams | NONE | NONE | 0.48409, 10 | 0.58187, 4 |
| 500 4-grams | NONE | NONE | NONE | 0.57400, 3 | **0.72125, 3** |
| 300 4-grams | NONE | NONE | NONE | 0.55450, 4 | **0.72937, 3** |
| 150 4-grams | NONE | NONE | NONE | 0.52630, 9 | 0.65093, 4 |
| 500 3-grams | 300 1-grams | NONE | NONE | 0.59750, 2 | 0.70031, 4 |
| 300 3-grams | 300 1-grams | NONE | NONE | 0.57139, 1 | 0.69218, 2 |
| 1000 3-grams | NONE | NONE | NONE | 0.59390, 2 | **0.70812, 10** |
| 1000 4-grams | NONE | NONE | NONE | 0.57480, 3 | **0.72125, 5** |
| 1000 3-, 4-, 5-grams | NONE | NONE | NONE | 0.56220, 2 | 0.69625, 3 |
| NONE | NONE | NONE | 20 1-, 2-, 3-grams | **0.62780, 2** | 0.64312, 10 |
| NONE | NONE | NONE | 10 1-, 2-, 3-grams | **0.61879, 3** | 0.549375, 8 |
| 10 2-, 3-, 4-grams | NONE | NONE | 10 1-, 2-, 3-grams | 0.58980, 3 | 0.69406, 8 |

Table 3: Results of different feature combinations with the Delta method using the Manhattan distance. The results consist of 2 numbers in the format $a, b$. $a$ corresponds to the accuracy obtained with the configuration and $b$ is the number of opponents that obtained that accuracy. Results that beat our baseline results are shown in bold.

**Euclidean Distance Extended Delta Method**

| Character n-grams | Word n-grams | POS-tag n-grams | Special character n-grams | PAN 2015 result | PAN 2013 result |
|---|---|---|---|---|---|
| 100 2-, 3-, 4-, 5-grams | 5 1-, 2-, 3-, 4-grams | 10 1-, 2-, 3-, 4-grams | 5 1-, 2-, 3-grams | 0.54570, 7 | 0.67750, 5 |
| 100 2-, 3-, 4-grams | 300 1-grams | NONE | NONE | 0.54080, 2 | **0.72125, 4** |
| 100 2-, 3-, 4-grams | NONE | NONE | 20 2-, 3-, 4-grams | 0.5618, 3 | 0.65812, 5 |
| 100 2-, 3-, 4-grams | NONE | 15 1-, 2-, 3-grams | 20 2-, 3-, 4-grams | 0.56500, 6 | 0.68031, 4 |
| NONE | NONE | 15 1-, 2-, 3-grams | NONE | 0.52110, 10 | 0.61156, 3 |
| 20 2-, 3-, 4-, 5-, 6-grams | NONE | NONE | NONE | 0.555, 3 | 0.66656, 2 |
| 150 2-, 3-grams | NONE | NONE | NONE | 0.55460, 3 | **0.70343, 9** |
| NONE | 50 2-, 3-grams | NONE | NONE | 0.52200, 9 | 0.56687, 3 |
| NONE | 20 2-, 3-grams | NONE | NONE | 0.48530, 9 | 0.58718, 3 |
| 500 4-grams | NONE | NONE | NONE | 0.55830, 3 | **0.70468, 8** |
| 300 4-grams | NONE | NONE | NONE | 0.52680, 3 | **0.71875, 6** |
| 150 4-grams | NONE | NONE | NONE | 0.53669, 6 | 0.64281, 7 |
| 500 3-grams | 300 1-grams | NONE | NONE | 0.5538, 2 | 0.69343, 6 |
| 300 3-grams | 300 1-grams | NONE | NONE | 0.53400, 2 | **0.71437, 6** |
| 1000 3-grams | NONE | NONE | NONE | 0.54910, 2 | **0.72156, 4** |
| 1000 4-grams | NONE | NONE | NONE | 0.57100, 2 | **0.72125, 3** |
| 1000 3-, 4-, 5-grams | NONE | NONE | NONE | 0.54579, 4 | **0.71968, 4** |
| NONE | NONE | NONE | 20 1-, 2-, 3-grams | **0.61200, 2** | 0.59375, 8 |
| NONE | NONE | NONE | 10 1-, 2-, 3-grams | **0.61370, 3** | 0.550625, 9 |
| 10 2-, 3-, 4-grams | NONE | NONE | 10 1-, 2-, 3-grams | 0.57980, 3 | 0.68125, 10 |

Table 4: Results of different feature combinations with the Delta method using the Euclidean distance. The results consist of 2 numbers in the format $a, b$. $a$ corresponds to the accuracy obtained with the configuration and $b$ is the number of opponents that obtained that accuracy. Results that beat our baseline results are shown in bold.

### 3.6.4 Author Specific SVM

In this approach we have experimented with a couple of different feature configurations. Configuration (A) consists of the 500 most frequent character-3, -4 and -5-grams, the 100 most frequent word-3 and -4-grams, the 20 most frequent postag-2, -3 and -4-grams. Configuration (B) consists of the frequencies of the 300 most frequent words. The most frequent n-grams are found in the Brown text corpus. We test only on the PAN 2013 dataset since the PAN 2015 dataset contains only a single known text per author and an SVM cannot train with a single datapoint in each class. We use the *sklearn* implementation of SVMs which internally use *libsvm*.

We use the RBF kernel and we choose hyperparameters via cross validation. Each configuration of features will use different hyperparameters. The cross validation is performed by looping through the list of authors. For each author we perform a grid search for value of $C \in \{10^{-2}, 10^{0}, \ldots, 10^{10}\}$ and $\gamma \in \{10^{-9}, 10^{-7}, \ldots, 10^{3}\}$. The best values for each author are found via leave one out cross validation. The final $C$ and $\gamma$ values are chosen as the configurations used most often by the authors. After we have found the best hyperparameters we run the classifier over all authors 100 times with those hyperparameters. The mean accuracy over the 100 runs is then computed.

Configuration (A) used the hyperparameters $C = 100$ and $\gamma = 0.00001$ and obtained an average accuracy of 0.84599 with a TPR of 0.95 and a TNR of 0.634. Configuration (B) used the hyperparameters $C = 100$ and $\gamma = 0.001$ and obtained an average accuracy of 0.84799 with a TPR of 0.99899 and a TNR of 0.632.

## 4 Results

As described earlier the PAN 2013 results are ranked using the F1 measure. The measure is defined using *precision* and *recall* which in PAN 2013 is defined as,

$$precision = \frac{correct\_answers}{answers} \tag{8}$$

$$recall = \frac{correct\_answers}{problems} \tag{9}$$

Since we answer all problems, *problems* and *answers* are the same in our case and therefore the F1 measure is the same as an accuracy,

$$F1 = 2\frac{precision \cdot recall}{precision + recall} = 2\frac{accuracy^2}{2accuracy} = \frac{2accuracy^2}{2accuracy} = \frac{accuracy^2}{accuracy} = accuracy. \tag{10}$$

Similarly as described earlier the PAN 2015 results are ranked using the product of AUROC and c@1. The AUROC is a measure of discrimination. That is, it measures the ability of a solution to distinguish between texts written by the same author and texts written by another author. An AUROC score is generally considered excellent when between 0.9 and 1, good when between 0.8 and 0.9, fair when between 0.7 and 0.8, poor when between 0.6 and 0.7 and a failure when between 0.5 and 0.6. The c@1 measure is chosen since it measures performance in the binary case. The c@1 measure does not use probabilities but classifies everything above 0.5 as a yes everything below as a no and a 0.5 as a don't know. Like F1 in PAN 2015, the c@1 also corresponds to an accuracy in our case since we answer all questions. The definition of c@1 is,

$$c@1 = \frac{1}{n}\left(n_c + \frac{n_u \cdot n_c}{n}\right) \tag{11}$$

| | PAN 2013 Dataset 1 | PAN 2013 Dataset 2 | PAN 2015 Dataset |
|---|---|---|---|
| **Accuracy** | 0.63191 | 0.61314 | 0.57850 |
| **TPR** | 0.51900 | 0.52492 | 0.54899 |
| **TNR** | 0.72408 | 0.70000 | 0.60800 |

Table 5: Result of running the delta method on two test sets included in PAN 2013 and single test set included in PAN 2015 with 4 opposing authors for the PAN 2013 set and 1 opposing author for PAN 2015.

where $n$ is the number of problems, $n_c$ is the number of correct answers and $n_u$ is the number of unanswered problems. So when $n_u$ is 0 we have,

$$c@1 = \frac{1}{n}\left(n_c + \frac{n_u \cdot n_c}{n}\right) = \frac{1}{n}\left(n_c + \frac{0 \cdot n_c}{n}\right) = \frac{1}{n}n_c = accuracy. \tag{12}$$

In this section we will describe how we have tested our solutions on the test datasets and give the results of those tests. When we are testing on the PAN 2013 dataset we will report an accuracy as that is the only performance measure. When we are testing on the PAN 2015 dataset we will report both the AUROC and the accuracy since both are used to measure performance.

### 4.1 Delta Method

The Delta method was tested by creating the same features for both the training dataset and the test datasets. We then computed the mean and standard variance of the training set and used that to normalize both the training and test dataset. For each text in the test dataset we then drew differing numbers of opposing texts from the training dataset. Those opposing texts were used as the opposition in the Delta Method. The number of opposing authors we used were the ones we found in the training section and were 4 for PAN 2013 and 1 for PAN 2015. The results for running the Delta Method on the two test sets included in PAN 2013 and one test set included in PAN 2015 is shown in Table 5. The Receiver Operating Characteristic (ROC) curve for the Delta Method is shown in Figure 1. It is created by computing the TPR and FPR for differing number of opposing authors.

The AUROC for the Delta Method on the PAN 2015 data was 0.59121. Resulting in a final score of $0.57850 \cdot 0.59121 = 0.34201$ for the PAN 2015 set.

### 4.2 Generalising Random Forest

In order to test the UBM approach proposed by [Pacheco et al., 2015], we start by computing our feature set and generating our UBM as described in Section 3.3.

The feature set and UBM, are then encoded according to Equation (6), and are used to train a random forest with parameters matching the ones used in experiments. That is all parameters being set to their default except `n_estimators` that is set to 1000. At this point we encode the feature set created from the test data, against the UBM, which is then fed to our trained random forest to get the predictions. This resulted in an accuracy of 0.60400, TPR of 0.63200 and TNR of 0.57600 on the pan 2015 dataset. On the other hand, we also chose to train our model using the alternate subtraction encoding, which doesn't make use of the UBM. The subtraction encoding under-performed relative to the UBM approach, with an accuracy of 0.58400 a TPR of 0.66800 and a TNR of 0.50000.

We have generated the ROC curve for both Random Forest tests. The AUROC was 0.63868 for the UBM method and 0.56821 for the Minus method. The curves is shown in Figure 2. This means that the Final Scores, of the methods are:

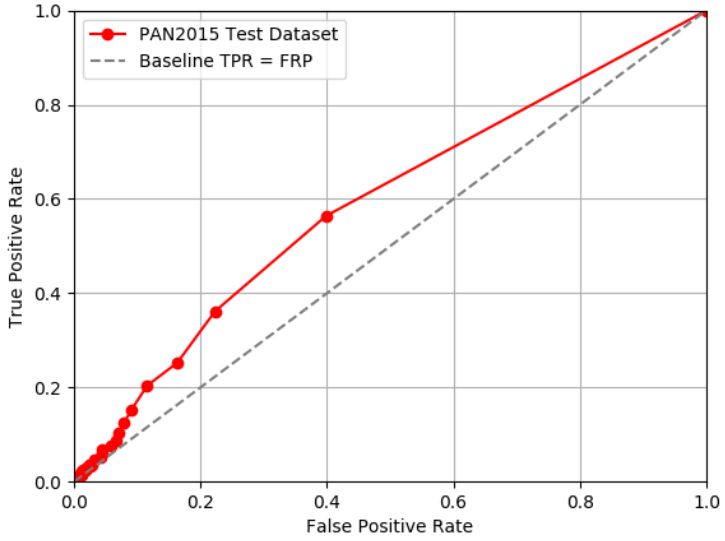



Figure 1: The ROC curve of the delta method with number of opposing authors varying from 0 to 100 using the test dataset for PAN 2015.

| | **PAN 2013 Dataset 1** | **PAN 2013 Dataset 2** | **PAN 2015 Dataset** |
|---|---|---|---|
| **Accuracy** | 0.72528 | 0.67291 | 0.61949 |
| **TPR** | 0.66750 | 0.75761 | 0.49760 |
| **TNR** | 0.77244 | 0.58953 | 0.74140 |

Table 6: Result of running the extended delta method on two test sets included in PAN 2013 and single test set included in PAN 2015 with the best configurations found in the training phase.

$$\text{Final Score UBM} = c@1 \cdot AUROC = 0.604 \cdot 0.63868 = 0.3858 \tag{13}$$

$$\text{Final Score Minus} = c@1 \cdot AUROC = 0.584 \cdot 0.56821 = 0.3318 \tag{14}$$

The ROC curve was generated by having the Random Forest give the probabilities for each unknown text belonging to the same author as the known. We then chose different thresholds between 0 and 1 (specifically $\{0.0, 0.01, \ldots, 1.0\}$) and computed the TPR and FPR for each of them. We then plotted those results as a function from FPR to TPR.

### 4.3 Extended Delta

The Extended Delta Method is tested by generating features according to the best configurations found in the training phase. Then the same procedure used to test the regular delta method is employed. The best configuration for the PAN 2013 data were the 100 most frequent character-2, -3 and 4-grams and the 300 most frequent words for 5 opposing authors. The best configuration for the PAN 2015 data were the 20 most frequent special-character-1, -2 and 3-grams for 2 opposing authors. The accuracies, TPRs and TNRs obtained on all test datasets are shown in Table 6. The ROC curve is shown in Figure 3 and the AUROC was 0.65188.

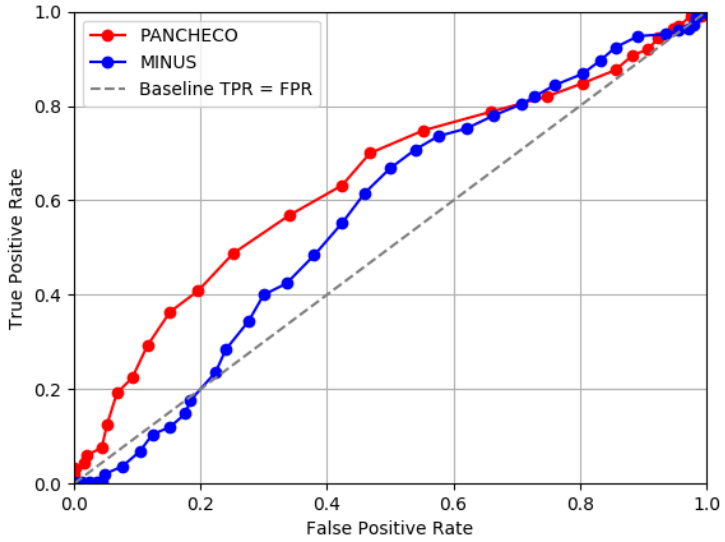



Figure 2: The ROC curve of the two Generalising Random Forest approaches.

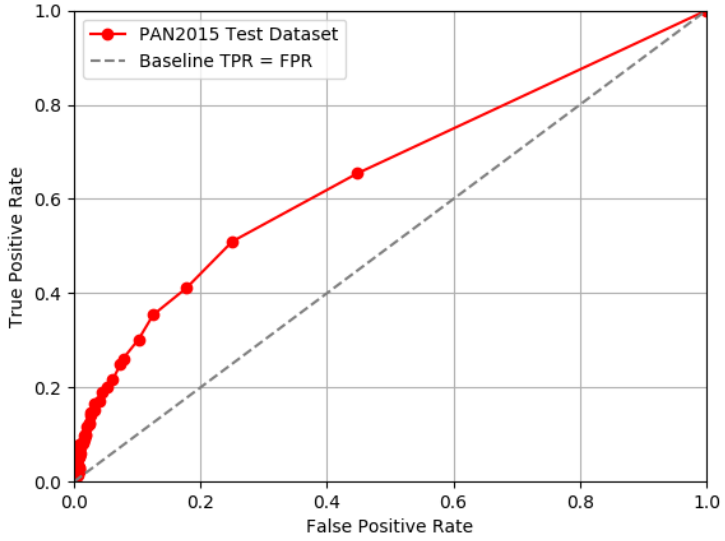



Figure 3: The ROC curve of the Extended Delta Method with number of opposing authors varying from 0 to 100 using the test dataset for for PAN 2015.

| | Test Dataset 1 | Test Dataset 2 |
|---|---|---|
| **Accuracy** | 0.77650 | 0.78066 |
| **TPR** | 0.71444 | 0.70785 |
| **TNR** | 0.82727 | 0.84437 |

Table 7: Results of the Author Specific SVM on the two test datasets for PAN 2013.

### 4.4 Author Specific SVM

The Author Specific SVM is tested by generating features for the training and test datasets on the PAN 2013 texts. For each author in the test dataset we extract features for all their known texts and their single unknown text. We then draw random texts from the training dataset which will serve as opponents to the texts written by the author. Then we train an SVM using the texts known to be written by the author and the texts from the training dataset and predict the unknown text using that SVM. The hyperparameters for the SVM is the ones we found best on the training dataset.

The best configuration on the training set was configuration B using the frequencies of the 300 most frequent words. The results on the two test datasets are shown in Table 7.

## 5 Discussion

### 5.1 Comparison to baseline approach

Our baseline approach was the Delta Method described in Section 3.2. One of the goals of this report was finding and implementing methods for authorship verification that could beat our baseline method. An overview of the different methods final results can be seen in Table 8. We were able to beat the baseline method both on the PAN 2013 dataset and the PAN 2015 dataset.

### 5.2 Comparison to other PAN results

The best results for the PAN 2013 dataset can be found in Appendix A and our results can be found in Table 8. We obtained the third best result out of 19 submitted results (excluding our own). The method we obtained the score with was the author specific SVM. The best approach that beat our method was the approach described by [Seidman, 2013]. They used the imposter method. To determine whether a document $X$ is written by the same author as has written $Y$, $X$ is compared to $Y$ and to a random set of imposter documents. If $X$ is found to be closer to $Y$ than the imposters it is reported as written by the same author and not otherwise. The approach is similar to ours as we also use text from other authors to do the verification. However [Seidman, 2013] did not use the training set as the set of imposters but rather generated the imposters from data found on the internet. It might be that the reason they performed better than we did was because they had a better set of imposters. [Seidman, 2013] also used predefined function words as features. A function word is a word that does not attribute meaning to a sentence but is used to build up the syntax of a sentence (words such as *do* in *we do not live here*). So instead of using only normal n-grams as we do they used prior knowledge about the language to construct good features.

The other PAN 2013 method that achieved better results than we did is described by [Veenman and Li, 2013]. Their method used the compression method of authorship verification. They determine the authorship of a text by generating a set of imposter texts and using the compression distance to see whether the unknown text is closer to the known texts or to the imposter set. Like [Seidman, 2013] the imposter set is generated by using external text and [Veenman and Li, 2013] even mention that the selection of the imposter set was an

| **Method** | | **PAN 2013** | **PAN 2015** |
|---|---|---|---|
| Delta (BASELINE) | Score | 0.62252 | 0.34201 |
| | TPR | 0.52196 | 0.54899 |
| | TNR | 0.71204 | 0.60800 |
| | Rank | 13/19 | 10/17 |
| Random Forest (UBM) | Score | n/a | 0.38576 |
| | TPR | n/a | 0.63200 |
| | TNR | n/a | 0.57600 |
| | Rank | n/a | 9/17 |
| Random Forest (Minus) | Score | n/a | 0.33183 |
| | TPR | n/a | 0.66800 |
| | TNR | n/a | 0.50000 |
| | Rank | n/a | 10/17 |
| Extended Delta | Score | 0.69909 | 0.40383 |
| | TPR | 0.71255 | 0.49760 |
| | TNR | 0.68098 | 0.74140 |
| | Rank | 10/19 | 8/17 |
| Author Specific SVM | Score | 0.77858 | n/a |
| | TPR | 0.71114 | n/a |
| | TNR | 0.83582 | n/a |
| | Rank | 3/19 | n/a |

Table 8: Final results for all our implemented algorithms on the test datasets of PAN 2013 and test dataset of PAN 2015. Since there are two test datasets for PAN 2013 *Score*, *TPR* and *TNR* in the PAN 2013 column are all an average over the values on the two datasets. The score in the PAN 2015 column is the product of the AUROC and the c@1 as described earlier. Algorithms that were only made for one of the dataset has *n/a* in the fields they are missing. The ranking is which rank we would have obtained in the PAN competitions had we competed in them.

| Feature Class | Feature | Importance Score |
|---|---|---|
| char-4-gram | 'for ' | 0.01230 |
| char-2-gram | 'in' | 0.01056 |
| char-5-gram | ' for ' | 0.00984 |
| char-3-gram | 'e o' | 0.00967 |
| special-char-2-gram | ',,' | 0.00887 |
| postag-3-gram | NOUN, ADP, DET | 0.00863 |
| char-5-gram | 'ould ' | 0.00800 |
| char-2-gram | 't ' | 0.00788 |
| postag-2-gram | NOUN, CONJ | 0.00739 |
| char-3-gram | 'for' | 0.00700 |
| char-3-gram | ' fo' | 0.00623 |
| postag-3-gram | DET, NOUN, PUNCT | 0.00582 |
| postag-2-gram | VERB, ADJ | 0.00563 |
| word-1-gram | 'for' | 0.00557 |
| postag-4-gram | VERB, DET, ADJ, NOUN | 0.00556 |
| char-4-gram | ' for' | 0.00538 |
| char-4-gram | '. Th' | 0.00499 |
| char-5-gram | 'here ' | 0.00496 |
| char-3-gram | 'ng ' | 0.00493 |
| char-3-gram | 'or ' | 0.00480 |

Table 9: The 20 features the random forest most often splits on when doing authorship verification, using the UBM Random Forest approach on the PAN 2015 dataset. The importance score represents how often the Random Forest splits on that particular feature.

important part of the solution. Again we might have been able to beat their result if we had tried using external texts as our set of opposing authors.

The best results for the PAN 2015 dataset can be found in Appendix B and our results can be found in Table 8. We obtained the 8'th place out of 17 submitted results (excluding our own). The method that we used was the Extended Delta Method. The Extended Delta Method is a distance based approach that does not involve any machine learning. It might be that the reason it performs so well on the PAN 2015 data is because of the lack of data in the PAN 2015 diminishes how well machine learning algorithms can perform. The best submission on the PAN 2015 task was described by [Bagnall, 2015]. The submission used a Recurrent Neural Network to make the prediction. The network was trained on the authors text and was supposed to predict the next character in a text given the context of the previous characters. A text was then classified as written by the same author by again using the imposter method. If the trained neural network was better at predicting the next character on an imposter than on the unknown text, the text was classified as not written by the same author. It was very surprising to us that the best performing method was a neural network as such networks normally require a lot more data to prevent overfitting.

### 5.3 Feature Importance

The *sklearn* random forest implementation has a build in notion of feature importance. The importance is estimated based on how often the forest splits on specific features. We have used the feature importance estimation to get an idea of which features are important for authorship verification. The 20 most important features according to the random forest are shown in Table 9.

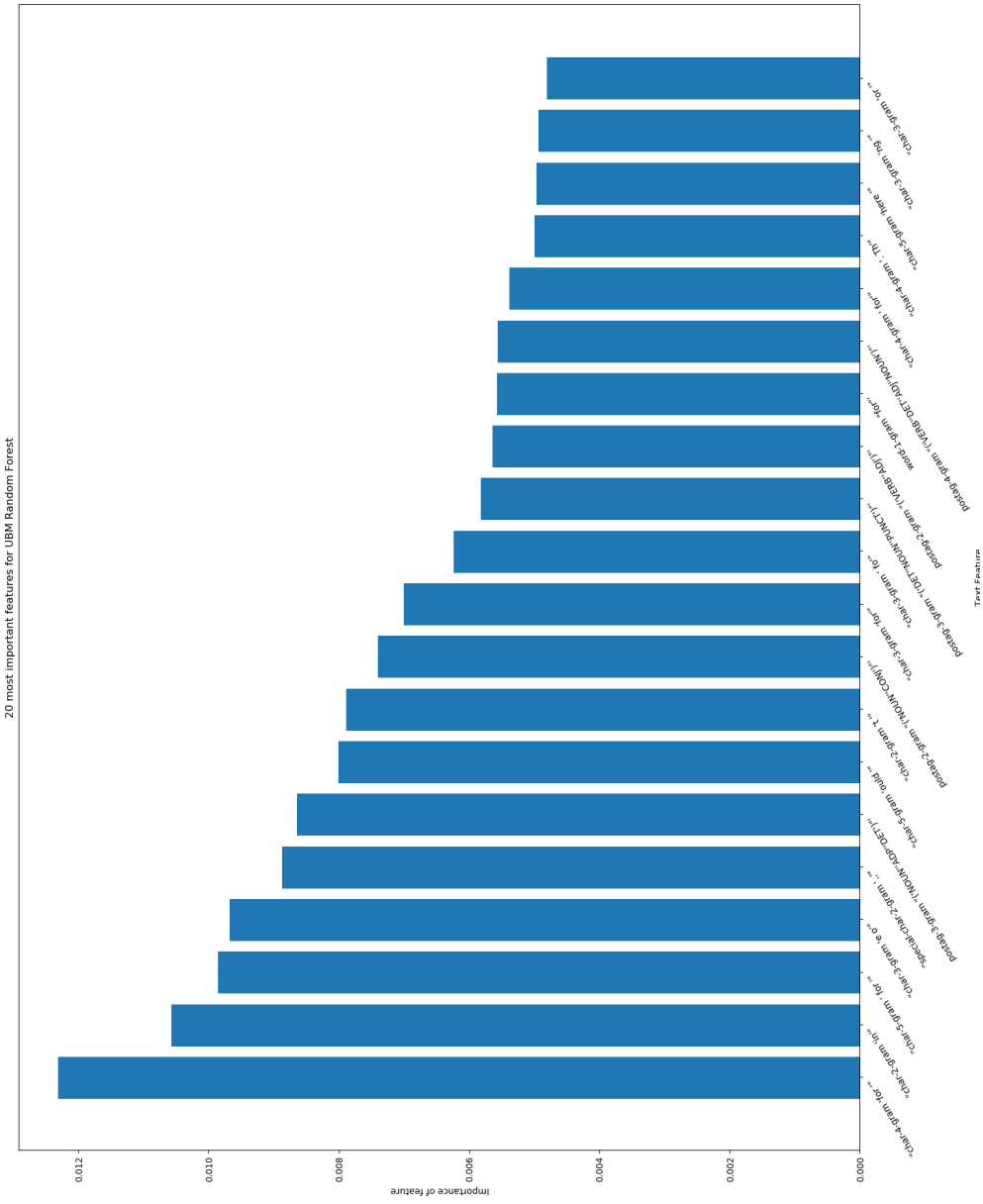



Figure 4: The 20 most important features after applying the UBM encoded features to the random forest model on the PAN 2015 dataset.

From that overview it is clear that the most important feature is the frequency of the word *for* as multiple features seem to capture that word. Most of the words captured by the different features seem to be English stop words. Both *for*, *in*, *could*, *should*, *would*, *here* and *or* seem to be represented in the list and are part of the list of English stop words [7]. Besides that an important feature is the special-character-2-gram consisting of two commas. That feature might capture an authors tendency to using long sentences. Longer sentences will more likely contain a larger amount of commas. The character-4-gram ". Th" most likely captures an authors tendency to start her sentences with the word "The". Besides that the POS-tags capture the general sentence structure of the authors and many POS-tags are important for the verification.

The reason the English stop words are so important is probably because they are consistently used over all genres. Texts about different topics and in different genres will probably all contain the words *the* and *for*. It is therefore easier to use them as an estimate of which author has written the text since they are not dependent on which genre or type of text the author was writing.

The least important features according to the random forest are shown in Table 10. All of those features consist of word-5-grams and all of them seem to be very specific to a source text. For example one of the features are the words "in the united states and". There is a good chance that no author will have that phrase at about the same frequency in all texts she writes. The phrase will clearly be more prevalent in texts about the United States and less prevalent in cooking recipes even though they have the same author.

It was not only word-5-grams that did not perform very well. After training our random forest model, it became apparent that both word-2-grams, word-3-grams and word-4-grams also does not perform very well. The reason for that might be the lack of text in the PAN 2015 dataset. The texts on PAN 2015 being limited to an average of 460 words meaning that there are few different word-n-grams in the texts. However, the word-n-grams extracted from the Brown corpus vary a lot and is usually very subject specific as described above. As such only a few of the word-n-grams extracted from the brown corpus are in the PAN 2015 texts. This only becomes more evident as n increases as the word-n-grams becomes more and more text specific.

Looking at Figure 5 however, we can see that the overall feature contribution is somewhat good. Based on the fact that 600 features were used, meaning that each feature importance would be $\frac{1}{600} = 0.00166$ if they were equally distributed. We have represented that level as a red line on the graph. As such rather than our model depending on a very small amount of features, it can be seen that over 50% of the feature set actually contributes to the classification, with an importance over the average. While this could very well only be the case for our selected features, it does lend some credence to the idea that using more features in the forest during training could increase accuracy.

### 5.4 Performance of our different methods

The Delta Method was our baseline method and has been found to perform well in the authorship verification setting [Evert et al., 2015]. On the PAN 2013 datasets we obtained an average accuracy of 0.62252 which is not a significant result, compared to the other results on the 2013 set. From the average TPR (0.52196) and TNR (0.71204) we can see that the main problem is the TPR. When the TPR is low we have a large number of FNs compared to TPs which means that we will often say a text is written by a different author even though it is written by the same author. On the training data we found that the number of opposing authors on the 2013 dataset should be 4 but it seems like the number was to high for the test dataset. Even though the results were not significant we still ranked 13/19 when comparing

[7] https://www.ranks.nl/stopwords

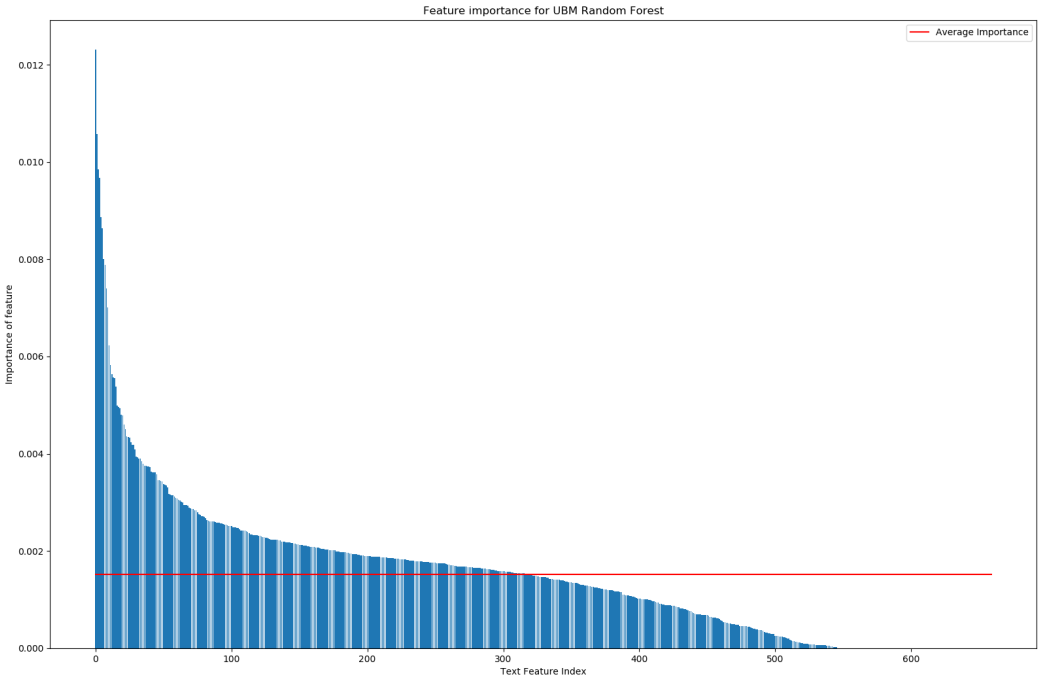



Figure 5: The feature importance for all UBM encoded features given to the random forest model on the PAN 2015 dataset.

| **Feature Class** | **Feature** | **Importance Score** |
|---|---|---|
| word-5-gram | is one of the most | 0.0 |
| word-5-gram | index words and electronic switches | 0.0 |
| word-5-gram | in the united states and | 0.0 |
| word-5-gram | at the foot of the | 0.0 |
| word-5-gram | we are born of god | 0.0 |
| word-5-gram | turned out to be a | 0.0 |
| word-5-gram | the state of rhode island | 0.0 |
| word-5-gram | the secretary of the interior | 0.0 |
| word-5-gram | the second half of the | 0.0 |
| word-5-gram | the first half of the | 0.0 |
| word-5-gram | the far end of the | 0.0 |
| word-5-gram | president of the united states | 0.0 |
| word-5-gram | of the state of rhode | 0.0 |
| word-5-gram | of the government of the | 0.0 |
| word-5-gram | if we are born of | 0.0 |
| word-5-gram | for the rest of the | 0.0 |
| word-5-gram | for a number of years | 0.0 |
| word-5-gram | at the time of the | 0.0 |
| word-5-gram | are born of god we | 0.0 |
| word-5-gram | year of our lord one | 0.0 |

Table 10: The 20 features the random forest least often split on when doing authorship verification on the PAN 2015 dataset. The importance score is how often the random forest splits on that feature.

to the other competitors that year. On the PAN 2015 dataset we obtained a final score of 0.34201 a TPR of 0.54899 and a TNR of 0.60800. Again the TNR is better than the TPR.

The Generalising Random Forest was off to a bad start start, as the paper it was based on [Pacheco et al., 2015] only got 6'th place in the PAN 2015 authorship verification task, and our implementation of that same method only ranked 9'th. The main focus of this method however, was the way it attempted to circumvent the lack of data, associated with each specific author, by instead learning on the known text dataset in its entirety. While the results of this attempt didn't land us at any place near the top, in terms of placement it did work. By comparing our UBM based results to the author specific Minus encoding we can see an improvement in terms of accuracy, thus making the UBM a viable choice when faced with small amounts of entry-specific data. We only ran the method on the PAN 2015 dataset where it got a score of 0.38576 for UBM and 0.33183 for Minus. Opposite to the Delta Method the main problem of the forest is not the TPR but the TNR. The rates obtained were TPR of 0.63200 for UBM and 0.66800 for Minus and TNR of 0.57600 and 0.50000 for Minus.

The Extended Delta Method achieved the best result of any of our methods on the PAN 2015 data. The ROC curve for the method is clearly the best one of any of our implemented methods. On the PAN 2013 dataset it used a collection of different features while on the PAN 2015 dataset the method used only 60 features and all of them were different special-character-n-grams. We tried the method using both the Euclidean and the Manhattan distance. They performed about the same but the Manhattan distance was a little better in all cases. Which is similar to the findings of [Evert et al., 2015]. In general Manhatten distances emphasise the differences over more small features instead of single large ones so it seems like it is important to consider the differences in many features when doing authorship verification. The method obtained an accuracy of 0.69909 on the PAN 2013 dataset and final score of 0.40383 on the PAN 2015 dataset. Both results are significant improvements over the original Delta Method we also tried. The TPRs were 0.71255 for PAN 2013 and 0.49760 for PAN 2015 and the TNRs were 0.68098 for PAN 2013 and 0.74140 for PAN 2015. It seems like the PAN 2013 method is good both at identifying when texts are written by the same author and when they are not while the PAN 2015 method is only good at identifying when texts are written by different authors.

The Author Specific SVM performed very well on the PAN 2013 dataset. As described by [Stamatatos, 2009] SVMs are great for doing Authorship Attribution and Verification since they are able to handle large amounts of features. And since representing a text document will typically use large vectors SVMs are a natural approach. Besides the accuracy the TPR and TNR achieved by the Author Specific SVM was also very good. However they were clearly better on the training dataset than on the testing dataset. The main drawback of the SVM method is that it requires either several examples of text from an author or a single long example that can be split into multiple. [Stamatatos, 2009] describes that performance for SVMs have previously been shown to drop significantly when only a small amount of text is available. Unfortunately the PAN 2015 dataset did not contain enough data for us to split the text into multiple and try the SVM approach on them. The accuracy, TPR and TNR obtained were 0.77858, 0.71114 and 0.83582. The results are excellent, the SVM are both good at finding when texts are written by the same author and when they are written by different authors.

### 5.5 Improvements

While the Random Forest method didn't provide any impressive results, the generalized approach might very well be useful in the future, in case one comes across another sparse dataset. An improvement could be to increase the number of features fed to the random forest. Not only quantity, but also different types of features than the ones used in this

report. This would work since the Random Forest selects the best features for classification. Thus we can, at the cost of runtime, train on a much larger feature-set and we suspect that this better results might be produced, a point supported by Figure 5 as mentioned earlier.

Similar to the Random Forest method the Author Specific SVM might also have been able to perform better by using more features. As explained earlier SVMs are very good at handling large amounts of features so we could have probably improved performance by adding more. Furthermore as described earlier the solutions that performed better than ours usually used extra text documents not from the dataset as opposing authors (imposters). We believe that we might have achieved better results if we had done that too.

We could have also tried more distance metrics/measures in the extended Delta Method. The Cosine Measure has by [Evert et al., 2015] been shown to perform better than the Manhattan distance. We could have also tried the Min/Max measure which [Aalykke, 2016] found to be the best distance measure for Danish texts.

# 6 Conclusion

We have presented a collection of machine learning and distance based solutions to the PAN 2013, and PAN 2015 tasks. We beat our baseline (the Delta Method) on both datasets. We implemented solutions that we applied to both datasets and solutions specifically made for one dataset.

We obtained third place in the PAN 2013 competition using our SVM solution. While our best results on the PAN 2015 set, the Extended Delta method, did beat the baseline as well, it didn't score as high on the PAN 2015 scoreboard. It scored 8'th in that year of the PAN competition.

We have experimented with several different types of features and we now have a good idea of which features better represents texts. Specifically stop-words and special characters have a far greater impact on authorship verification, than for example content specific words.

We have experimented with how the amount of data you have available influences the problem of authorship verification. The PAN 2013 dataset had many more words available per author and the accuracies we were able to obtain reflected that fact. Both of the solutions that performed better than our solution on the PAN 2013 dataset used extra texts besides the ones given to train their solutions. That supports our suspicion that the amount of data there is available is the greatest predictor of the final score of a solution.

# 7 Future Work

Two of the algorithm we have implemented has only been applied to one of the two datasets we have worked with. In particular the Random Forest approach has only been applied to the PAN 2015 data and the SVM approach has only been applied to the PAN 2013 data. The main reason for that, is that the Random Forest approach requires a single known text per author and the SVM approach require multiple known texts per author. It would be interesting to apply the algorithms to the opposite datasets and look at their performance there. We can transform the dataset with only a single known text to a dataset with multiple by splitting the known text into a collection of known texts. Similarly we can transform the dataset with multiple known texts to a dataset with a single known text by concatenating the known texts. By making those two transformations we could have run all methods on all data and found results for all of them.

We would also like to apply our different implemented methods to larger datasets. We have found in this assignment that having more text available per author improves the performance of our methods. All our implemented methods performed better on the PAN 2013 dataset than on the PAN 2015 dataset. For example there is a Danish company, MaCom,

that produce software for turning in and managing school assignments. MaCom has a large database containing several texts for each student and they have an interest in authorship verification. They specifically want to verify that assignments uploaded by students match their previous assignments and are not written by someone else. MaCom's main requirement is a method that has a high TPR. The reason is that they don't want to falsely accuse anyone of not having written their own assignment, while it doesn't matter as much if they miss some assignment that is written by someone else. When the TPR is high it means that there is few FNs and many TPs. FNs are as described earlier when we say a text is written by a different author while it is actually written by the same.

Since MaCom has much data available per student their case most closely match that of the PAN 2013 dataset. The method we implemented with the best TPR on the PAN 2013 dataset was our Extended Delta Method approach. So it would be interesting to apply that approach to MaCom's dataset. The SVM approach also had a very promising TPR on the training dataset so it would also be interesting to try that.

## References

[Aalykke, 2016] Aalykke, A. H. (2016). Computational authorship attribution in danish high schools. Master's thesis, DTU.

[Angelova et al., 2015] Angelova, G., Bontcheva, K., and Mitkov, R., editors (2015). *Recent Advances in Natural Language Processing, RANLP 2015, 7-9 September, 2015, Hissar, Bulgaria*. RANLP 2015 Organising Committee / ACL.

[Bagnall, 2015] Bagnall, D. (2015). Author Identification using multi-headed Recurrent Neural Networks—Notebook for PAN at CLEF 2015. In [Stamatatos et al., 2015].

[Bartoli et al., 2015] Bartoli, A., Dagri, A., De Lorenzo, A., Medvet, E., and Tarlao, F. (2015). An Author Verification Approach Based on Differential Features—Notebook for PAN at CLEF 2015. In [Stamatatos et al., 2015].

[Castro et al., 2015a] Castro, D., Adame, Y., Pelaez, M., and Muñoz, R. (2015a). Authorship Verification, Combining Linguistic Features and Different Similarity Functions—Notebook for PAN at CLEF 2015. In [Stamatatos et al., 2015].

[Castro et al., 2015b] Castro, D. C., Arcia, Y. A., Brioso, M. P., and Guillena, R. M. (2015b). Authorship verification, average similarity analysis. In [Angelova et al., 2015], pages 84–90.

[Efstathios Stamatatos et al., 2013] Efstathios Stamatatos, W. D., Verhoeven, B., Potthast, M., Stein, B., Juola, P., Sanchez-perez, M. A., and Barrón-cedeño, A. (2013). E.: Overview of the author identification task at pan-2013. In *Notebook Papers of CLEF 2013 LABs and Workshops (CLEF-2013) (2013*.

[Evert et al., 2015] Evert, S., Proisl, T., Jannidis, F., Pielström, S., Schöch, C., and Vitt, T. (2015). Towards a better understanding of burrows's delta in literary authorship attribution. Denver, Colorado, USA. NAACLHLT 2015.

[Gómez-Adorno et al., 2015] Gómez-Adorno, H., Sidorov, G., Pinto, D., and Markov, I. (2015). A Graph Based Authorship Identification Approach—Notebook for PAN at CLEF 2015. In [Stamatatos et al., 2015].

[Gutierrez et al., 2015] Gutierrez, J., Casillas, J., Ledesma, P., Fuentes, G., and Meza, I. (2015). Homotopy Based Classification for Author Verification Task—Notebook for PAN at CLEF 2015. In [Stamatatos et al., 2015].

[Hansen et al., 2014] Hansen, N., Lioma, C., Larsen, B., and Alstrup, S. (2014). *Temporal context for authorship attribution: a study of Danish secondary schools*, pages 22–40. Springer.

[Kuppili, 2015] Kuppili, A. (2015). What is the time complexity of a random forest, both building the model and classification?

[Layton, 2014] Layton, R. (2014). A simple Local n-gram Ensemble for Authorship Verification—Notebook for PAN at CLEF 2014. In [Rangel et al., 2014].

[Maitra et al., 2015] Maitra, P., Ghosh, S., and Das, D. (2015). Authorship Verification - An Approach based on Random Forest—Notebook for PAN at CLEF 2015. In [Stamatatos et al., 2015].

[Pacheco et al., 2015] Pacheco, M., Fernandes, K., and Porco, A. (2015). Random Forest with Increased Generalization: A Universal Background Approach for Authorship Verification—Notebook for PAN at CLEF 2015. In [Stamatatos et al., 2015].

[Peñas and Rodrigo, 2011] Peñas, A. and Rodrigo, A. (2011). A simple measure to assess non-response. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, pages 1415–1424, Stroudsburg, PA, USA. Association for Computational Linguistics.

[Posadas-Durán et al., 2015] Posadas-Durán, J.-P., Sidorov, G., Batyrshin, I., and Mirasol-Meléndez, E. (2015). Author Verification Using Syntactic N-grams—Notebook for PAN at CLEF 2015. In [Stamatatos et al., 2015].

[Rangel et al., 2014] Rangel, F., Rosso, P., Chugur, I., Potthast, M., Trenkmann, M., Stein, B., Verhoeven, B., and Daelemans, W. (2014). Overview of the 2nd author profiling task at pan 2014. In *CLEF 2014 Evaluation Labs and Workshop – Working Notes Papers*, Sheffield, UK.

[Seidman, 2013] Seidman, S. (2013). Authorship Verification Using the Impostors Method—Notebook for PAN at CLEF 2013. In [Efstathios Stamatatos et al., 2013].

[Stamatatos, 2009] Stamatatos, E. (2009). A survey of modern authorship attribution methods. *J. Am. Soc. Inf. Sci. Technol.*, 60(3):538–556.

[Stamatatos et al., 2015] Stamatatos, E., Daelemans, W., Verhoeven, B., Juola, P., López-López, A., Potthast, M., and Stein, B. (2015). Overview of the author identification task at pan 2015. In *CLEF 2015 Evaluation Labs and Workshop – Working Notes Papers*, Toulouse, France. CEUR, CEUR.

[Veenman and Li, 2013] Veenman, C. and Li, Z. (2013). Authorship Verification with Compression Features. In [Efstathios Stamatatos et al., 2013].

# A Pan 2013 Results

The results of the PAN 2013 Authorship verification task, on English texts. Most of the participants has, like us, answered all problems and therefore has the same Precision, Recall and as a result the same F1.

| Author | F1 | Precision | Recall |
|---|---|---|---|
| Seidman. | 0.8 | 0.8 | 0.8 |
| Veenman & Li. | 0.8 | 0.8 | 0.8 |
| Layton et al. | 0.767 | 0.767 | 0.767 |
| Moreau & Vogel. | 0.767 | 0.767 | 0.767 |
| Jankowska et al. | 0.733 | 0.733 | 0.733 |
| Vilariño et al. | 0.733 | 0.733 | 0.733 |
| Halvani et al. | 0.700 | 0.700 | 0.700 |
| Feng & Hirst. | 0.700 | 0.700 | 0.700 |
| Ghaeini. | 0.760 | 0.760 | 0.760 |
| Petmanson. | 0.667 | 0.667 | 0.667 |
| Bobicev. | 0.644 | 0.655 | 0.633 |
| Sorin | 0.633 | 0.633 | 0.633 |
| van Dam. | 0.600 | 0.600 | 0.600 |
| Jayapal & Goswami. | 0.600 | 0.600 | 0.600 |
| Kern. | 0.533 | 0.533 | 0.533 |
| BASELINE | 0.500 | 0.500 | 0.500 |
| Vartapetiance & Gillam. | 0.500 | 0.500 | 0.500 |
| Ledesma et al. | 0.467 | 0.467 | 0.467 |
| Grozea | 0.400 | 0.400 | 0.400 |

# B Pan 2015 Results

The results of the PAN 2015 Authorship verification task, on English texts. Empty fields are a result of the authors no specifying these parameters clearly in their papers.

| Author | Final Score | c@1 | AUC |
|---|---|---|---|
| Douglas Bagnall. | 0.61 | 0.76 | 0.81 |
| Daniel Castro et al. | 0.52041 | 0.694 | 0.74987 |
| Josue Gutierrez1 et al. | 0.51 | 0.69 | 0.74 |
| Mirco Kocher et al. | 0.5082 | 0.6890 | 0.7375 |
| Erwan Moreau et al. | 0.453 | . | . |
| María Leonor Pacheco et al. | 0.43811 | 0.57429 | 0.76287 |
| Manuela Hürlimann et al. | 0.41 | . | . |
| Juan Pablo Posadas Durán et al. | 0.39999 | 0.58800 | 0.68025 |
| Promita Maitra et al. | 0.34749 | 0.57732 | 0.60174 |
| Alberto Bartoli et al. | 0.323 | 0.56 | 0.578 |
| Helena Gómez Adorno et al. | 0.2809 | 0.53 | 0.53 |
| Stanimir Nikolov et al. | 0.2582 | 0.5243 | 0.4926 |
| Julián Solórzano et al. | 0.258 | 0.5 | 0.517 |
| Oliver Pimas et al. | 0.2565 | 0.506 | 0.50692 |
| Yunita Sari et al. | 0.20055 | 0.5 | 0.4011 |
| Oren Halvani et al. | . | 0.675 | . |
| Seifeddine Mechtik et al. | . | 0.59 | . |