https://doi.org/10.35662/unine-thesis-3027

Automatic Author Profiling and Verification

PhD Thesis presented to the Faculty of Sciences
Institute of Computer Sciences
University of Neuchâtel

For the degree of Doctor of Science.
by

Ikae Catherine Omal

Approved by the dissertation committee:

Prof. Jacques Savoy, Thesis director
University of Neuchâtel, Switzerland

Prof. Elöd Egyed-Zsigmond, National Institute of Applied Sciences of Lyon (INSA Lyon), France
Dr. Valerio Schiavoni, University of Neuchâtel, Switzerland

Defended on November 24, 2022

unine
Université de Neuchâtel
Faculté des sciences

Rue Emile-Argand 11
CH-2000 Neuchâtel
doctorat.sciences@unine.ch

IMPRIMATUR POUR THESE DE DOCTORAT

La Faculté des sciences de l'Université de Neuchâtel autorise l'impression de la présente thèse soutenue par

Madame Catherine IKAE

Titre :

"Automatic Author Profiling and Verification"

sur le rapport des membres du jury composé comme suit:

- Prof. Jacques Savoy, directeur de thèse, Université de Neuchâtel, Suisse
- Dr Valerio Schiavoni, Université de Neuchâtel, Suisse
- Prof. Elöd Egyed-Zsigmond, LIRIS, INSA, Lyon, France

Neuchâtel, le 28 novembre 2022

Le Doyen, Prof. R. Bshary

Abstract

This thesis mostly discusses the style-based text categorization problem, where the objective is to identify the author's demographics, such as gender, age range, and language variety, based on a set of texts. Also to determine whether two writings (chat, threatening e-mail, doubtful testimony, essays, text messages, business memos, fanfiction texts) were authored by the same person by contrasting the writing styles of the two texts by applying the vector difference text representation. We also create a stable and straightforward paradigm for feature reduction iteratively. This reduction will result to a more explainable decision.

We begin by assessing the effectiveness of several machine learning models using the complete vocabulary. The two-step feature selection technique is then used to design a feature reduction strategy. After testing the models with these reduced features, we were able to examine how the performance variation would appear in the two scenarios. We went on to test further feature reduction by applying χ^2 and PMI scoring functions to select the top 300 features.

With the use of several CLEF-PAN datasets, we test our models, and we can see that Extra Trees, Random Forest, or Gradient Boost often produce the best results. Furthermore, empirical evidence reveals that the feature set can be effectively condensed using χ^2 and PMI scoring methods to about 300 features without compromising performance. Additionally, we can see that by discarding non informative features, decreasing the text feature representation not only cuts down on runtime but also improves performance in some cases.

With the difference vector text representation approach we demonstrate how utilization of confidence-based approaches can benefit classification accuracy in the author verification. We can see that small differences in vectorial representation indicates higher similarity, but documents with a large differences are not authored by the same writer. Several performance measures are obtained including accuracy, area under the curve (AUC), c@1 and Final Score (FS). Our research shows a strong correlation between all performance with measurements FS and AUC having the strongest correlation. We take into account only the accuracy to draw conclusion about the different text representation methods. Our experiments therefore show that the best scoring model include TFIDF feature set since it considers both occurrence frequency and the distribution of terms across the collection.

Résumé

Cette thèse s'intéresse principalement aux problèmes de classification de textes fondée sur le style dont le but est d'identifier les caractéristiques de l'auteur comme son âge, sexe, son idiolecte, en se basant sur un ensemble de ses écrits. De plus, on aborde la question de savoir si deux textes (comme des chats, courriels menaçants, testaments douteux, essais, mémos, ou fictions) ont été écrits par la même personne en comparant leur style d'écriture selon différentes représentations. Nous proposons un processus de sélection des attributs simple et stable. Cette réduction nous conduit à proposer une décision possédant un plus grand pouvoir explicatif.

Nous débutons ce travail par analyser l'efficacité de plusieurs modèles basés sur l'apprentissage automatique et recourant à l'ensemble du vocabulaire. Une procédure de réduction des attributs en deux étapes peut alors être appliquée. Nous pouvons alors comparer les performances de divers modèles avec des réduction du nombre d'attributs basés sur notre approche, le 2 ou le PMI. Dans tous les cas, le nombre d'attributs est réduit à 300.

Sur la base de la collection de documents de différentes campagnes d'évaluation CLEF-PAN, nous avons testé notre approche avec plusieurs baselines. On constate que les modèles Extra Trees, Random Forest, ou Gradient Boost produisent souvent les meilleures performances. De plus, la réduction des attributs au nombre de 300 permet d'obtenir des performances similaires. Cette diminution permet également de réduire la taille des représentations des documents et donc de réduire le temps de calcul. Parfois, nous observons même un gain d'efficience.

Dans le cadre de la vérification d'auteur, et selon diverses représentations des textes, nous pouvons également améliorer la qualité des résultats. Ainsi, les documents présentant de grandes différences de représentation ne sont pas écrits par la même personne. Dans ce but nous avons appliqué différentes mesures de performance (AUC, c@1, Final Score (FS)) dont les résultats sont corrélés en particulier AUC et FS. En tenant compte uniquement du taux de réussite, la pondération TFIDF offre les meilleures performances.

Acknowledgement

I'm extremely grateful to my Supervisor, Prof. Jacques Savoy for his invaluable patience, feedback and who generously provided knowledge and expertise during my PhD thesis.

Likewise, I would like to thank the members of the jury, namely Prof. Elöd Egyed-Zsigmond (Institut national des sciences appliquées de Lyon) and Dr. Valerio Schiavoni (Université de Neuchâtel), for the time they devoted to evaluating this dissertation.

I am also grateful to my colleagues in the department, for their kindness, friendship and moral support.

I should not forget to acknowledge my family. Their confidence in me has sustained my enthusiasm and upbeat attitude throughout this process.

Contents

1 Introduction

The Internet has established itself as a large and interactive platform for communication that enables the sharing of knowledge among users of various demographics, including gender, age, socioeconomic status, and location. Thanks to several services that allow for the simple sharing of information, such as SMS, chats, blogs, tweets, and Facebook posts, among others, social media has recently experienced a significant increase in popularity. This suggests that there are a lot of fresh texts and images being shared by authors who, for the most part, we don't know anything about.

This publicly accessible information raises crucial security concerns, particularly in light of the rise in the availability of pseudonymous posts, chats, threatening emails, or anonymously created documents online. Such accounts have been used to perform illegal or deceptive acts such as sexual harassment and extortion. To detect and prevent this type of illicit acts, the discipline is known as forensic linguistics, makes use of linguistic knowledge to study texts that evidence this type of illegal behaviour [28] [62]. On the other hand, companies want to know or automatically extract demographic information from their customers reports, e-mails or complains. Having a better knowledge about their customers is a prime importance for on line shops or more generally information providers.

As text, one can consider a novel, a play, a speech, a set of tweets, a customer report, an e-mail, a passage inside a larger message. For most of our investigation, we are working with text written in English but other natural languages are possible.

There are a number of reasons why it is crucial to understand some pertinent information about members of social networks. For instance, marketing professionals are interested in learning the identities and demographic details of different consumers in order to better target their advertisements [12]. Politicians want to identify some personal information about their supporters.

Because of these, there is a growing demand to analyze user profiles on social media. Given that it would be impossible to conduct such a study manually, computational tools must be used to conduct this analysis automatically. This leads to mainly two questions. Can we extract some demographic information about the author of a message? Second can we identify or verify the identity of the author of a written report?

1.1 Motivation and Objectives

The general domain of this thesis is text categorization, a field in linguistics and machine learning are the two most important sub domain. One can be more precise and specify that quantitative view of the language is the essential part. Thus corpus linguistics provide tools and corpora to evaluate the quantity of the proposed solutions.

From a machine learning domain, our focus is on text-based models with the challenge to select or generate the best text representation to fulfil our goal. In this line one can consider content based approaches or style-based models. Usually however, the best models tend to combine both representations. In this thesis, we will also consider the problem to automatically select the best subset of features able to produce the best answer. Moreover, trying to reduce the features set to a few hundred terms will simplify the search of a good explanation of the proposed decision. As a second main question with machine learning models is to chose the best matching between representation of a query text and learned ones.

1.2 Authorship Analysis

Authorship analysis is a growing field in data mining that uses linguistic techniques to provide insight about the author. Everything in the modern world is accessible online, which encourages criminal and evil activities. Thus, it is now necessary to identify some general information about the author of a message. Even if could be completed with images, photos or videos, we want to limit our investigation to text-based content.

Authorship analysis uses of linguistic techniques to clarify who wrote a work in issue. It can be used, for example, to identify the text's most likely author from a sample of suspects or the most likely demographic information about an anonymous author.

Authorship analysis is focusing on style and not the content of the message to determine who wrote a document. This is based on the presumption that the author of a text has a distinctive writing style that presents personal lexical and syntactical choices distinct from other writers [22].

The term "stylometry" refers to techniques for authorship analysis that quantifies stylistic qualities and looks for textual features that are similar within a class but different between classes to determine who the genuine author of a text is. Aspects of linguistic style used in stylometry include word or sentence length distribution, vocabulary richness, and various frequencies (such as of words, word lengths, word forms, characters, or combinations of them). As stylistic fingerprints, stylometric traits are utilized to identify the author of unidentified or contested works.

1.2.1 Author Profiling

The author profiling task (AP) involves taking demographic information. These information could be, gender, age range, place, profession, socio-economic status, or native tongue [79] [136] [105]. Efforts have also been made to determine other aspects such as which authors are bots or humans [32], whether the text was written by many authors (and to define the number of possible authors), the author of a given text is keen to be a spreader of fake news [144], an author of a text spreads hate speech [39], whether the author of a given text spreads irony and stereotypes [36].

In order to identify these characteristics, formal materials like books, newspapers, or magazines were evaluated under the assumption that for each text we know the corresponding category (or gender). The challenge of figuring out a person's profile by looking at their social media accounts, however, has gained momentum recently [119] [115] [121].

Style-based techniques which involve examining the author's writing where the subject of the two main approaches namely style-based or content-based models. Such strategies have been most successful in tackling the issue of AP in social networks. The primary contribution of various works is dependent on the choice of characteristics that can gauge the author's style and subject matter as seen in most of the PAN-CLEF tasks [119] [115] [121].

To categorize difficult writing styles, the studies on author profiling have provided a set of stylistic traits. Every feature category has a place in predicting the authors' demographic characteristics. Combinations of these characteristics have also been used to categorize writing styles of authors. The three most common methods for capturing an author's viewpoint are style based on characters, style-based on vocabulary, and style-based on syntax.

Understanding the authors writing style will enable us answer questions such as: Do men and women write similarly, or are there significant differences between the two [60]? What characteristics best distinguish typing between different age groups [119] [115]? Is it feasible to identify someone's

personality traits with any degree of accuracy from a text excerpt? Can we also identify the characteristics that distinguish texts from one another among many language varieties [120]? Can we identify the author's identity as a person or a bot [117]? Can we determine from the writing style if the author is eager to distribute false information [116]? Can we determine if a text's author spreads hate speech based on how it is written [14]?

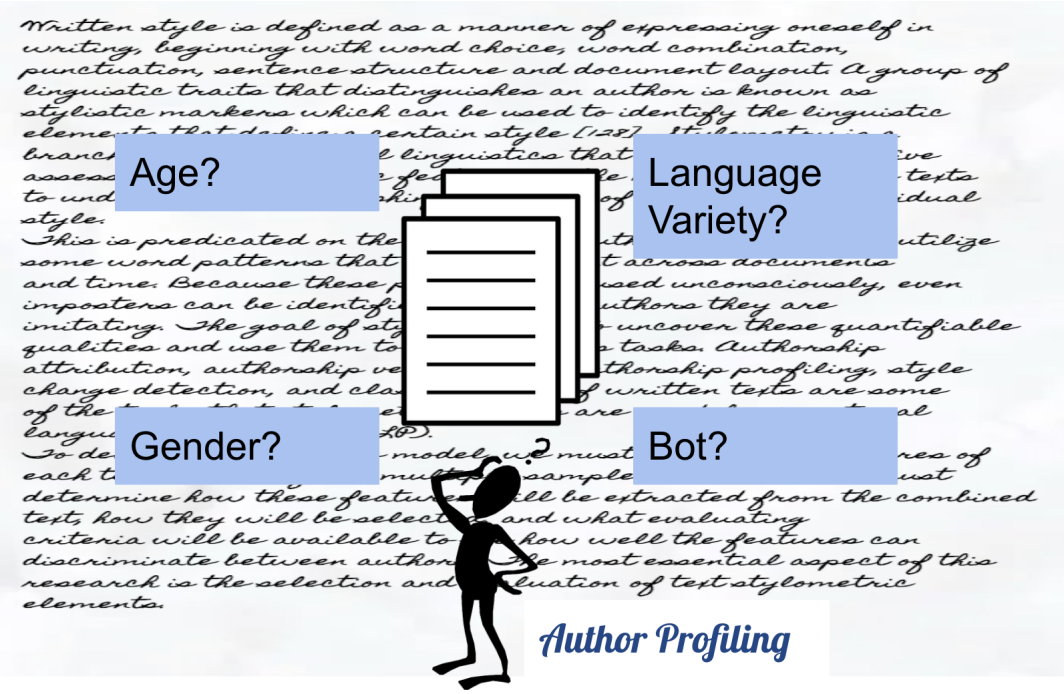


Figure 1: A text used to extract an author profile (gender, age, diversity of languages spoken, etc.)

1.2.2 Authorship Verification

Authorship verification is determining whether two writings (chat, threatening e-mail, doubtful testimony, essays, text messages, business memos, fanfiction texts) were authored by the same author by contrasting the writing styles of the two texts.

Author verification is a similarity detection challenge of answering a query. Given a sample text, was the text produced by a specific person? The problem of identifying the author of a disputed document given a list of potential authors and samples of their writing can also be understood in this way.

The difficult part, though, is to disprove shared authorship and determine an appropriate threshold for similarity when we can still confirm the shared ancestry of two texts. Additionally, the system should produce a likelihood (or degree of credibility) that the suggested author is the genuine one. In addition the system must provide support for its decision by giving some reasons supporting the result, rather than being restricted to a single binary conclusion. Lastly, it would be preferable to indicate that there is insufficient evidence to make a determination rather than risk giving the incorrect response [72].

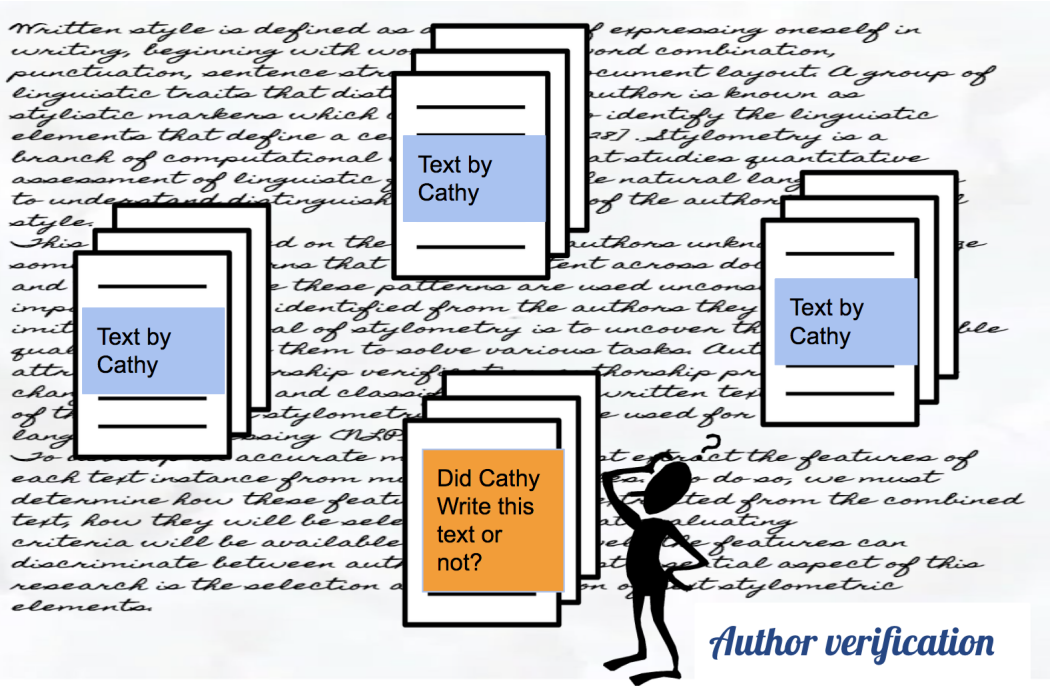


Figure 2: The claimed authorship of the unidentified text must be established or refuted.

2 Stylometry

Written style is defined as a manner of expressing oneself in writing, beginning with word choice, word combination, punctuation, sentence structure and document layout. A group of linguistic traits that distinguishes an author is known as stylistic markers which can be used to identify the linguistic elements that define a certain style [132] . Stylometry is a branch of computational linguistics that studies quantitative assessment of linguistic features in the natural language texts to understand distinguishing features of the author's individual style.

This is predicated on the notion that authors unknowingly utilize some word patterns that are consistent across documents and time. Because these patterns are used unconsciously, even imposters can be identified from the authors they are imitating. The goal of stylometry is to uncover these quantifiable qualities and use them to solve various tasks. Authorship attribution, authorship verification, authorship profiling, style change detection, and classification of written texts are some of the tasks that stylometry studies are used for in natural language processing (NLP).

To develop an accurate model, we must extract the features of each text instance from multiple samples. To do so, we must determine how these features will be extracted from the combined text, how they will be selected, and what evaluating criteria will be available to see how well the features can discriminate between authors. The most essential aspect of this research is the selection and evaluation of text stylometric elements.

In this chapter, we review some general information regarding the most effective features studied in literature (character based features, lexical based features, syntax based features, content based features) to discriminate between different authors. To achieve this, we consider different independent feature-scoring selection functions (information gain, gain ratio, pointwise mutual information, odds ratio, chi-square, GSS, TFIDF). To find the predictive relationship between the inputs and outputs, we will examine groups of machine learning algorithms (tree-based methods, neural network inspired methods, instance-based, dimensionality reduction, regression, bayesian, ensemble and deep learning based algorithms).

2.1 Data Pre-processing and Style based Feature Extraction

Text pre-processing is the process of converting a document into a format that is easy to analyze for your needs. Stop words, punctuation marks, special characters, letters normalization, numerals, and other characters can all be found in the text documents. The following are some examples of preprocessing techniques:

- Lower casing: changing all upper case letters to lower case. This is applicable to most tasks and is applicable to most applications;
- Stemming is the process of reducing inflection in words (e.g. troubled, troubles) to their root form (e.g. trouble). Stemming chops off the ends of words (suffices) in the hope of correctly transforming words into its root form;
- Lemmatization removes inflections and map a word to its root form (or entry in the dictionary).
- Normalization is the process of transforming a text into a canonical (standard) form. For example, the word "gooood" and "gud" can be transformed to "good";
- Noise removal is about removing characters, digits and pieces of text that can interfere with your text analysis;

- Stopword removal (task dependent) is the removal of commonly used words in a language such as “a”, “the”, “is”, “are” for the English language:
- Expand contractions such as “don‘t” to “do not” and “can‘t” to “can not” helps to standardize text:
- Tokenization is the process by which a large quantity of text is divided into smaller parts called word-tokens or in short tokens. These tokens make up the features from which selection can be made. "i am the king." is turned into "i", "am", "the", "king", ".".

Stylometry investigates features related to the distribution of words, the use of punctuation, the grammar, the structure of the sentence or paragraph and so on. Typically, the set of features to be analysed in a text are divided into the following categories:

2.1.1 Character based Features

A text is a sequence of characters (or letters), and character tokenization can result into various character-based features such as: all characters, special characters, white-space characters, capital letters, lower case letters, and numeric data in the text, n-gram of characters, character frequencies.

Features from characters in text have been described in terms of their number of occurrence: The total number of character n-grams, the number of capitalized letters, the frequency of special characters. In addition, one can consider their proportions: the proportion of capital letters to total characters, the proportion of white-space characters to total characters, the number of tab spaces divided by the total number of characters (known as the tab space ratio), the proportion of white to non-white spaces, the proportion of capital to lower case letters and the ratio of numeric values to the total number of characters.

2.1.2 Lexical based Features

Lexical features describe the set of words that an individual uses. Depending on the tokenization used, various lexical features can be extracted some of which include: used hapax legomena (i.e., words occurring once) and hapax dislegomena (the number of words that occur twice), n-grams of words (n consecutive words). In addition, features could also be selected if they contain positive/ negative emotional words. In some cases, features may include counts of acronyms, foreign words, capitalized words, and words with digits. Statistical count of words such as the average length of words used, the average of words used per sentence leading to the an analysis of the vocabulary richness of an author.

2.1.3 Syntax based Features

A syntax is the set of rules, principles, and processes that govern the structure of sentences in a given language. For some authors, the syntax can be limited to the order or arrangement of words and phrases to form proper sentences. Function words having little lexical meaning express grammatical relationships among other words within a sentence. These words help in defining the relationships between the elements of a sentence. They are also the most common words found in any text. Unfortunately, given the length of a tweet, such features do not contribute significantly to the representation of such texts on their own. Function words include closed part of speeches (POS), such as pronouns (she, they), determiners (the, that), prepositions (in, of), auxiliary verbs (be, have), modal verbs (may, could), conjunctions (and, but) and quantifiers (some, both).

As additional features, one can consider POS tags (e.g nouns, adverbs) extracted from the sentence syntax and also n-grams of such POS tags.

2.1.4 Content based Features

The themes that users write about and discuss on social media are referred to as content features. For example when writing a review of a product or service, the author employs emotive terms. The text may contain words that express positive or negative emotion, rage, or religion-related subjects. It is also possible to detect hate-related content. When it comes to a corpus taken from forums or topic-specific sources, the frequency of these specific keywords in the text is extremely revealing and has been used to detect terrorism and cyber-paedophilia.

Each of the characteristic types described above is critical in text analysis. They all play a role in distinguishing different authors of texts, text genre or time period of writing. Because taking account of all those features can generate lots of predictors, we'll need a process to select only the features with the most discriminating power to represent the given text body.

2.2 Feature Selection

It is known that high dimensionality of the features appearing in document representation is a real problem for many machine learning algorithms. Document vectors are sparse resulting into a document vector containing only few entries different from zero. Reduction of the features used for the representation of documents is required while ensuring its relevance in the text classification by making sure that only non-informative and noisy features are excluded.

Selection of each feature according to their discriminating power over the others requires the measure of the fitness of each attribute over each category. One can define a discriminating power for each attribute. Then the features are ranked, from the most appropriate to the less suitable. The proposed subset is simply determined by extracting the top m most discriminative features or those having a global score larger than a predefined threshold.

To measure the discriminating power of a term t_i according to a given category (or author) c_j (with $j = 1, 2, \ldots, r$) a contingency table can be generated as depicted in Table 1. The value a indicates the number of texts belonging to the category $c_j$in which the term t_i occurs. When considering all other classes (denoted by $\bar{c}_j$), the term t_i appears in b other texts. Thus, in the whole corpus, this term occurs in $a + b$ documents, while we can count $a + c$ texts labelled with the category c_j .

	Category c_j	Category $\bar{c}_j$	Total
Term t_i	a	b	$a + b$
Term $\bar{t}_i$	c	d	$c + d$
Total	$a + c$	$b + d$	$n = a+b+c+d$

Table 1: Contingency table

To determine the discriminating power of a feature for a given category, various scoring functions have been suggested.

2.2.1 Odds Ratio (OR)

Odds Ratio compares the probability of a feature existing in one category with the probability for it existing in the other. It gives a positive score to features that appear more often in one category than in the other, and a negative score if it appears more in the other. A score of zero means the probability for a feature to appear in one category is exactly the same as the probability for it to appear in the other [57] [131].

Let $P(t|c)$ be the probability of a randomly chosen word being t, given that the document belongs to the class c. Then the odd is defined as $\frac{P(t|c)}{1-P(t|c)}$ and the odds ratio will be expressed as a ratio of the two odds:

$$\begin{aligned} OR(t_i, c_j) &= \frac{\frac{P(t_i|c_j)}{1-P(t_i|c_j)}}{\frac{P(t_i|\bar{c}_j)}{1-P(t_i|\bar{c}_j)}} = \frac{(P(t_i|c_j)) * (1-P(t_i|\bar{c}_j))}{(1-P(t_i|c_j)) * (P(t_i|\bar{c}_j))} \\ &= \frac{\frac{a}{a+c} * \left(1 - \frac{b}{b+d}\right)}{\left(1 - \frac{a}{a+c}\right) * \frac{b}{b+d}} = \frac{a*d}{b*c} \end{aligned} \tag{1}$$

2.2.2 Information Gain (IG)

This function measures the amount of information obtained for category prediction by knowing the presence or absence of a term in a document. Those terms whose information gain is less than some predetermined threshold are removed from the feature space [131]. IG is derived from entropy measuring the homogeneity of a distribution. IG(c, t) of an attribute t relative to a collection of dataset c, is defined as

$$\begin{aligned} IG(t_i, c_j) &= \sum_{c \in \{c_j, \bar{c}_j\}} \sum_{t \in \{t_i, \bar{t}_i\}} p(t,c) * log_2 \left(\frac{p(t,c)}{p(t) * p(c)}\right) \\ &= \frac{a}{n} * log_2 \left(\frac{a*n}{(a+b)*(a+c)}\right) + \frac{b}{n} * log_2 \left(\frac{b*n}{(a+b)*(b+d)}\right) \\ &+ \frac{c}{n} * log_2 \left(\frac{c*n}{(a+c)*(c+d)}\right) + \frac{d}{n} * log_2 \left(\frac{d*n}{(b+d)*(c+d)}\right) \end{aligned} \tag{2}$$

The value returned by this function is large if a positive association exists between the term t and the category c. A small positive value signifies the absence of a discriminative power for the term t and the category c.

2.2.3 Gain Ratio (GR)

GR is an extension to IG, expressed as a ratio of information gain to the intrinsic information. Gain ratio is usually applied when defining the node in a decision tree. In this context, GR takes number and size of branches into account when choosing an attribute there by reducing its bias on high-branch

attributes [131]. Returning a positive value to signal either a positive or negative association between the term t and the category c. Independence is indicated by a value close to 0.

$$\begin{aligned} GR(t_i, c_j) &= p(t_i, c_j) * log_2\left(\frac{p(t_i, c_j)}{p(t_i) * p(c_j)}\right) + p(\bar{t}_i, c_j) * log_2\left(\frac{p(\bar{t}_i, c_j)}{p(\bar{t}_i) * p(c_j)}\right) \\ &= \frac{a}{n} * log_2\left(\frac{a * n}{(a+b) * (a+c)}\right) + \frac{c}{n} * log_2\left(\frac{c * n}{(a+c) * (c+d)}\right) \end{aligned} \tag{3}$$

2.2.4 Pointwise Mutual Information (PMI)

PMI measures how much information a term contains about a class. It measures how much information the presence/absence of a term contributes to making the correct classification decision. The magnitude of PMI will indicate if an association between a feature and a category exist or not [30] [131]. PMI can therefore be applied to decide if a feature is informative or not, and a feature selection is done on that basis. Having less features often improves the performance of your classification algorithm. To calculate PMI, as a ratio the joint probability ($p(t_i, c_j)$) and the probability of occurrence of term t_i multiplied by the probability of selecting a text belonging to the category c_j.

$$\begin{aligned} PMI(t_i, c_j) = log_2\left(\frac{p(t_i, c_j)}{p(t_i) * p(c_j)}\right) &= log_2\left(\frac{\frac{a}{n}}{\frac{a+b}{n} * \frac{a+c}{n}}\right) \\ &= log_2\left(\frac{a * n}{(a+b) * (a+c)}\right) \end{aligned} \tag{4}$$

2.2.5 GSS (Galavotti-Sebastiani-Simi)

GSS signals a positive association with a positive value, and an opposition with a negative value. When the returned value is close to 0, there is no relationship between the feature and the corresponding category [131].

$$\begin{aligned} GSS(t_i, c_j) &= (p(t_i, c_j) * p(\bar{t}_i, \bar{c}_j)) - (p(t_i, \bar{c}_j) * p(\bar{t}_i, c_j)) \\ &= \frac{(a * d) - (b * c)}{n^2} \end{aligned} \tag{5}$$

2.2.6 Chi-square (χ^2)

This function is used to test the independence of two variables in the correct case, the independence of a feature and a category is provided by a χ^2 value. The higher value of the χ^2, the closer relationship the variables have [131]

$$\begin{aligned} \chi^2(t_i, c_j) &= \frac{n * ((p(t_i, c_j) * p(\bar{t}_i, \bar{c}_j))(p(t_i, \bar{c}_j) * p(\bar{t}_i, c_j)))^2}{p(t_i) * p(\bar{t}_i) * p(c_j) * p(\bar{c}_j)} \\ &= \frac{n * (a * d - c * b)^2}{(a+c) * (b+d) * (a+b) * (c+d)} \end{aligned} \tag{6}$$

2.2.7 Term Frequency–Inverse Document Frequency (TFIDF)

TFIDF is a statistical measure used in information retrieval and text mining that quantifies the importance of a word in a document by evaluating how relevant a word is to a given document in a collection of documents [129] [112].

$$TFIDF = Term\ Frequency(TF) * Inverse\ Document\ Frequency(IDF)$$

Term frequency (TF) corresponds to the frequency of a word in a document. This value highly depends on the length of the document. If t is term (word), d is document (sequence of words), N is count of corpus, and $corpus$ is the total document set, then

$$TF(t, d) = count\ of\ t\ in\ d\ /\ number\ of\ words\ in\ d$$

Document Frequency (DF) is the count of occurrences of term t in the document set N. DF is the number of documents in which the word is present.

$$DF(t) = number\ of\ documents\ where\ t\ occurs$$

Inverse Document Frequency(IDF) is the inverse of the document frequency which measures the informativeness of term t.

$$IDF(t) = log(N/(DF + 1))$$

$$TFIDF = TF(t, d) * log(N/(DF + 1)) \tag{7}$$

2.3 Feature Vector Representations

Since machine learning models can only process numerical values therefore, tokens, subwords, characters, must be converted into numerical formats. Some of the common text feature vector representations used in deep learning are discussed in the following paragraphs.

One-Hot Encoding: When a categorical feature is encountered in a data collection, one-hot encoding may be the best option. It works by substituting each category with a vector full of zeros, except for the location of its corresponding index value, which has a value of 1. The different one-hot vectors of documents are changed, and a given phrase is turned into a 2D-matrix with the shape (n, m), where n is the number of tokens in the sentence and m is the vocabulary size.

Count Vectorizer: Based on frequencies, a count vectorizer can compress an entire sentence into a single vector. As *One-Hot Encoding*, each position of a count vector is allocated to a certain token, and its value denotes how many times that token appears in the document. Tokens are created from the corpus initially, and then a vocabulary is created to map the tokens to their corresponding ids. Rather than creating a separate vector for each token, a count vector simply counts the number of times each token appears in a sentence and assigns that number to the appropriate position in the vector.

Word2vec takes a large corpus of texts as input and outputs a vector space with hundreds of dimensions in which each feature/token in the corpus is allocated to a corresponding vector w_i in the space. As a result, once the word vectors have been computed words in the corpus with similar contexts are clustered together in the space [91].

Difference vector: for the author profiling tasks, we created a word representation comparable to the *Count Vectorizer* in our studies. However, the value assigned to each token in each document is divided by the document length. As a result, the token value is unaffected by document length discrepancies.

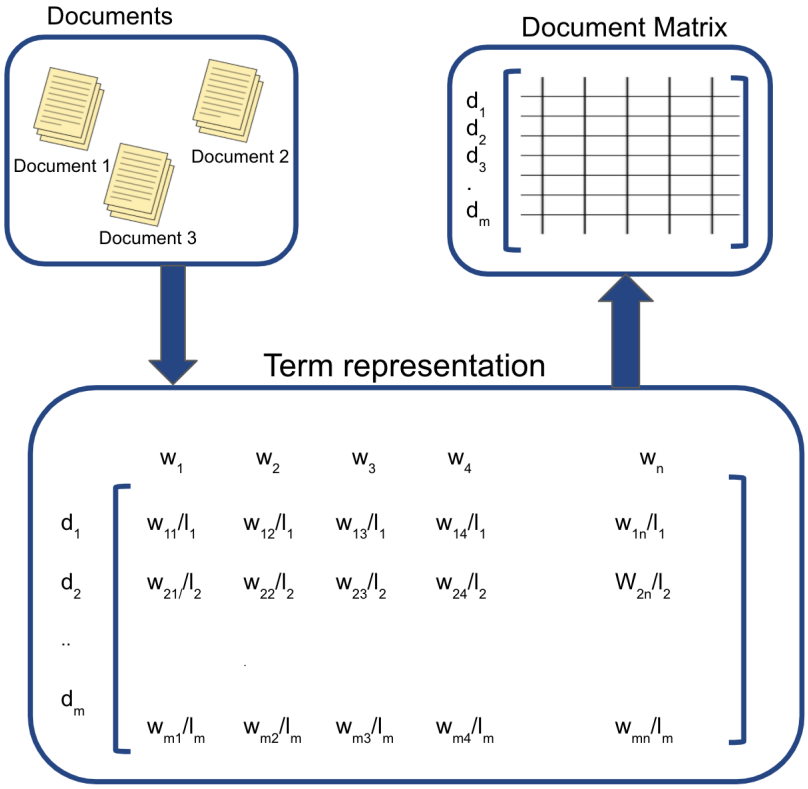


Figure 3: Term representation for a document

We employed the vector difference of the terms in the author verification tasks. The resulting document vector is made up of vector discrepancies between the values of the individual terms in each of the document pairings. We can tell that a vector with a small difference indicates better similarity, but documents with a large vector difference are not authored by the same writer.

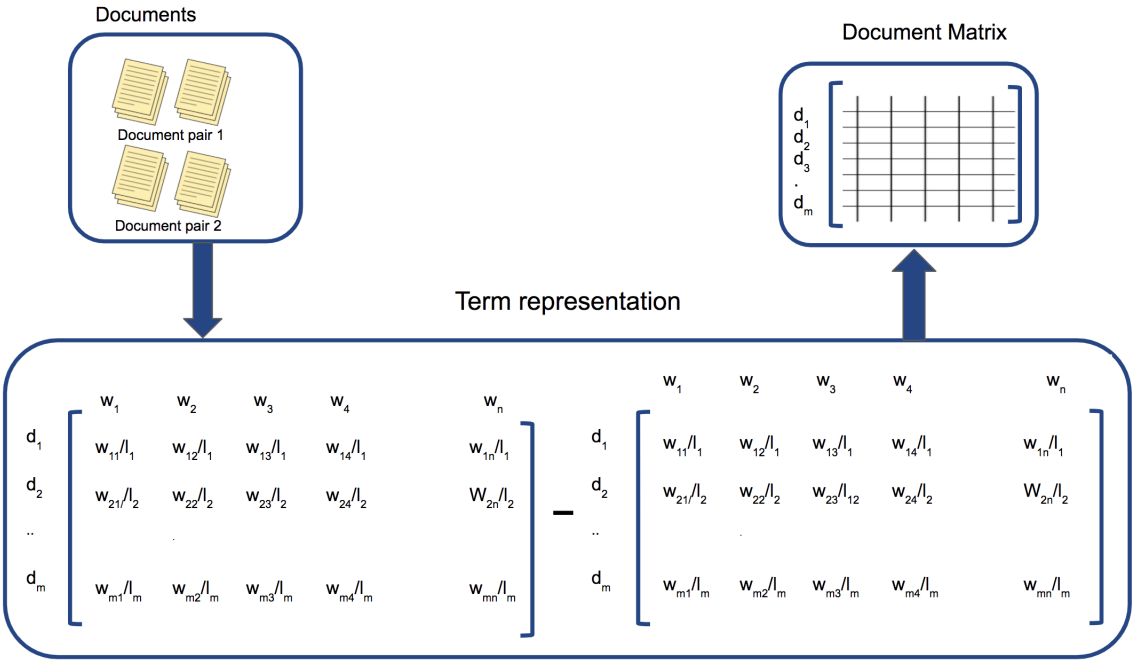


Figure 4: Term representation of document pairs

2.4 Machine Learning Algorithms

Approaches to computational stylometry rely heavily on machine learning methods for supervised and unsupervised learning of the feature space. Most research utilize support vectors machines (SVM), K-nearest neighbours (k-NN), principal component analysis (PCA), decision trees all of these are listed in detail in [132].

The fact that data from real life is not generated randomly, there always exist patterns in it, although we don't know exactly what they are. Machine learning models have parameters that can be fitted and optimized to be able to learn the patterns which is represented by features or descriptors so as to decide to which categories an unseen query text belongs. This is done by exploring the patterns of the given training data and make prediction for the new data set.

Text classification consists in constructing a vector from the text characteristics to feed a machine learning algorithm to determine the profile of the author through pattern recognition.

Machine learning algorithms in NLP systems generally experiment with various combinations of lexical, syntactic and semantic features to identify the most effective feature set. The most frequently used classifies are described in the following section.

2.4.1 Instance-based Algorithms or k-Nearest Neighbours (k-NN)

Instance-based learning (sometimes called memory-based learning[1]) performs classification by comparing new problem instances with instances seen in training. Complexity of such algorithms grow with the data. in the worst case, if n is the size of the training items and m the number of features, the computational complexity of classifying a single new instance is O(n*m).

k-Nearest Neighbours (k-NN) is a simple supervised classifier algorithm, that creates a decision surface according to both the distribution of the target value and the features that adapts to the shape

of the data distribution. k-NN was developed to perform discriminant analysis when reliable parametric estimates of probability densities are unknown or difficult to determine [66]. The algorithm relies on distance for classification and so assumes that similar items exist in close proximity of the k-neighbours.

The main problem in this approach is to determine an efficient similarity measure or (distance) for identifying neighbours of a particular document. Traditional k-NN uses Euclidean as a distance function. Other variants of the distance function are demonstrated in [78] but with no conclusion as to which is the best distance function. For proper comparison of the distance measures, the data must be represented with a similar format to enable effective distance computation. So the measurements must be all in the same standardized units.

The second challenge of k-NN is the classification time is long and it is difficult to find optimal value of k. A small value of k could be too sensitive to noise while a large value of k may include majority points from the other class.

k-NN has been used in different aspects of text classification as for example: [76] to classify short texts such as Twitter messages, blogs, chat massages, book and movie summaries, forums, news feeds, and customer review. [97] built a k-NN classifier for email classification where incoming emails, are each considered as a single document using TFIDF as the features. For type (bot vs. human) and the gender (male vs. female) prediction on twitter text data, [59] used k-NN algorithm achieving accuracy of (bot vs. human) 0.894, (male vs. female) 0.80.

2.4.2 Support Vector Machines (SVM)

This learning scheme is based on the idea of finding a hyperplane that best divides a dataset into two classes assuming that the feature space is linearly separable. The idea of classification with SVM is to find the best line in two dimensions or the best hyperplane in more than two dimensions in order to help us separate our space into classes. Support vectors are the data points that lie closest to the decision surface (or hyperplane). They are the data points most difficult to classify. They have direct bearing on the optimum location of the decision surface. SVM algorithm can operate even in fairly large feature sets as the goal is to define the largest margin of separation of the two classes data rather than having a perfect classification.

[67] introduces support vector machines for text categorization using words ranked according to their information gain. [11] identifies a reduction in feature set for automatic classification of news texts using SVM. [128] explores SVM and its variants for multi-category news classification using TFIDF features. [93] applies SVM for sentimental analysis on Epinions.com movie review data. [33] showed how to use SVM in authorship analysis showing that SVM achieves higher accuracies than the other classifiers on full word forms and 'tagwords', which are a combination of grammatical tags and function words. [140] described an automatic process for adjusting the threshold of the SVM to relax the conservative nature of SVMs used in text classification. [111] deployed SVM for text classification of Twitter text for a weather information system. [123] sentiment analysis for identification of ontologies of pre diabetes uses the SVM algorithm with term frequency as a feature extraction method.

2.4.3 Decision Tree Algorithms

Decision tree-based algorithms serve as the fundamental step in application of the decision tree method. Tree based algorithms are considered to be one of the best and mostly used supervised learning methods since they map non-linear relationships quite well.

- **Decision Tree** is a tree-structured classifier, where internal nodes represent the features of a dataset, branches represent the decision rules and each leaf node represents the outcome. In a decision tree, there are two node types, namely decision node and leaf node [126]. Decision nodes are used to make any decision and have multiple branches one for each answer. Leaf nodes are the output of those decisions and do not contain any further branches. The classifications are performed on the basis of features of the given dataset.

  [98] proposes the use of TFIDF features and decision tree technique for classification of documents. [149] carried out sentiment analysis on a large collection of customers' reviews on Amazon products. [6] uses TFIDF feature weighting method for fake news classification of fake news.

- **Extra Trees (Extremely Randomized Trees)** is an ensemble machine learning algorithm which works by creating a large number of unpruned decision trees from the training dataset. Predictions are made by averaging the prediction of the decision trees in the case of regression or using majority voting in the case of classification [43]. Extra Trees algorithm will randomly sample the features at each split point of a decision tree then fits each decision tree on the whole. The algorithm depends on the following hyper-parameters: the number of decision trees in the ensemble, the number of input features to randomly select and consider for each split point, and the minimum number of samples required in a node to create a new split point which must be optimised.

  [141] uses meta heuristics-based feature selection methods and employs Extra Trees classifier to classify emails into spam and ham with 95.3% accuracy out performing decision trees and random forest.

2.4.4 Bayesian Algorithms

Naive Bayes are a group of supervised machine learning classification algorithms based on the Bayes theorem. Bayes theorem can be used to calculate conditional probability applied in machine learning. These classifiers assume that the value of a particular feature is independent of the value of any other feature [54].

$$P(C_i \mid p) = \frac{P(C_i) * P(p \mid C_i)}{P(p)} \tag{8}$$

where:

- P(C_i) = is the prior probability of class C_i (target viable),
- P(p) = is the probability of the predictor p (attribute / independent variable),
- P(C_i|p) = is the posterior probability of class C_i given the predictor p,
- P(p|C_i) = is the likelihood, which is the probability of the predictor p given class C_i.

The denominator is a constant since all of the values of the attributes p are known. Combining the above equation taking into account multiple predictors, the equation for classification becomes:

$$P(C_i \mid p) \propto P(C_i) * P(p_1|C_i) * P(p_2 \mid C_i) * P(p_3 \mid C_i) * \ldots * P(p_n|C_i) \tag{9}$$

$$P(C_i \mid p) = P(C_i) * \prod_{k=1}^{m} P(p_k \mid C_i) \tag{10}$$

Therefore, the class label predicted by the model is the one with the highest probability.

- **Gaussian Naive Bayes** is a variant of Naive Bayes for continuous attributes assuming that the feature follow Gaussian normal distribution. The likelihood is assumed to be:

$$P(p_k \mid C_i) = \frac{1}{\sqrt{2\pi\sigma_i^2}} \exp\left(-\frac{(p_k - \mu_i)^2}{2\sigma_i^2}\right) \tag{11}$$

  Gaussian Naïve Bayes was used for the early detection of breast and lung cancers [71]. [92] applies TFIDF based text features and Extra Trees classifier for feature selection together with Gaussian Naïve Bayes for fake news detection in Bengali.

- **Bernoulli Naive Bayes** is a variant of Naive Bayes for for Boolean attributes. The attribute p_k can only take values of 1 or 0 (true or false).

$$P(p_k \mid C_i) = P(k \mid C_i)p_k + (1 - P(k \mid C_i))(1 - p_k) \tag{12}$$

  If p_k is 1, conditional probability result into $P(k \mid C_i)$ or if p_k is 0 the the probability is $1 - P(k \mid C_i)$

  [145] uses Bernoulli Naive Bayes for fake news detection.

- **Multinomial Naive Bayes** consider a feature vector where a given term represents the number of times it appears (frequency) suitable for discrete features (e.g., word counts for text classification).

$$P(p_k \mid C_i) = \frac{(\sum_{k=1}^{n} p_k)!}{\prod_{k=1}^{n} p_k!} \prod_{k=1}^{n} p_{ik}^{p_k} \tag{13}$$

  This variant of Naive Bayes has been used in several fields of text classification such as: movie reviews Sentiment Analysis (SA) [1], emotion detection of Twitter post using unigram with POS tag [113]; spam filtering [69].

2.4.5 Artificial Neural Network Algorithms

Neural network (NN) is a learning scheme based on the mimic of the human brain. During the learning stage, NN compares the value delivered by the output unit of each neuron with actual value in order to adjust the weights of all neurons to improve the prediction.

- **Multilayer Perceptrons (MLP)** is a type of of feed forward (data flows in the forward direction from input to output layer) neural network. Such a network consists of three types of layers the input layer, output layer and hidden layer. The input layer receives the input signal to be processed for prediction and classification which is performed by the output layer. An arbitrary number of hidden layers that are placed in between the input and output layer are the computational engine of the MLP [17].

  [3] uses a Multilayer Perceptron (MLP) based ensemble learning for financial sentiment analysis from microblogs and news headlines datasets. Naive Bayes algorithm (NB) and Multilayer Perceptron (MLP) network are combined with hybrid system called NB-MLP for Arabic sentiment classification [5]. Multilayer Perceptron (MLP) is applied in movie reviews classification by exploiting Decision Tree-based feature ranking [68].

- **Stochastic Gradient Descent** optimises an objective function equipped with the parameters of a model and updates parameters for each training sample [20]. It is an optimization algorithm for minimizing the loss of a predictive model with regard to a training dataset. Gradient Descent finds the set of input variables for a target function that results in a minimum value of the

target function, called the minimum of the function. Gradient descent involves calculating the gradient of the target function with respect to the specific values of the input values. A hyper-parameter called step size (learning rate) is used to scale the gradient and control how much to change each input variable with respect to the gradient. For stochastic gradient descent, the target function that is being minimized is loss function. The optimisation process uses as functions such as momentum, root mean squared ropagation (RMSProp) and adaptive movement estimation (Adam) which can be tuned.

[31] improves the performance of SGD for text classification by by fine-tuning hyper-parameters. In [142] SGD classifier with unigram and bigram TFIDF features is used for classification of suspicious Bengali text.

2.4.6 Dimensionality Reduction Algorithms

Dimensionality reduction seeks a lower-dimensional representation of numerical input data that preserves the salient relationships in the data. There are many different dimensionality reduction algorithms and no single method works for all datasets. Below, we discuss two dimensionality reduction algorithms:

- **Principal Component Analysis (PCA)** is a linear dimensionality reduction technique that can be utilized for extracting information from a high-dimensional space (features that represent the data) by projecting it into a lower-dimensional sub-space [56] usually 2D. It tries to preserve the essential parts that have more variation of the data and remove the non-essential parts with fewer variation. It takes a dataset having many features, and simplifies that dataset by selecting a few principal components from original features. The main idea of dimensionality reduction, is to speed up machine learning algorithm training and testing time considering the data has a lot of features and and a slow ML algorithm. PCA focuses on capturing the direction of maximum variation in the dataset.

  [65] uses word and character for authorship attribution with PCA. For fake new detection, [139] uses TF-IDF, countvectorizer and n-gram features of the review content and then principal component analysis to reduce the feature. PCA showed best performance when applied to reviews from Amazon.

- **Linear Discriminant Analysis (LDA)** is a linear machine learning algorithm LDA that attempts to find a feature subspace which maximizes class separability [155]. The model seeks to find a linear combination of input variables that achieves the maximum separation for samples between classes (class centroids or means) and the minimum separation of samples within each class.

  [4] discusses LDA dimensionality reduction method for Arabic text classification using Euclidean distance measure. [151] presents an application of linear discriminant analysis (LDA) to document classification.

2.4.7 Ensemble Algorithms

Ensemble is an approach to machine learning that seeks better predictive performance by combining the predictions from multiple models. Ensembles tend to yield better results when there is a significant diversity among the models.

- **Random Forest** consists of a large number of individual decision trees that operate as an ensemble. Each individual tree in the random forest spits out a class prediction and the class with

the most votes becomes our model prediction [23]. This model can perform well given a large feature sets because it combines the predictions of various decision trees to build a more robust classifier. While constructing new decision trees, this method uses a random subset of features which gets rid of spurious features and improving the robustness of our estimate.

[21] combined data enrichment with the introduction of semantics in random forest to improve short text classification. [90] described a new method on random forest and feature selection (FS) for text classification. [77] performs sentiment classification of Youtube comments using the random forest, and Word2Vec Skip-gram for features extraction. [37] explores random forest with several term weighting method for sentiment analysis in Indonesian language.

- **AdaBoost (Adaptive Boosting)** technique that aims at combining multiple weak classifiers to build one strong classifier. The underlying idea is to assume that a single classifier may not be able to accurately predict the class of an object. Moreover, when multiple weak classifiers are used with each one progressively learning from the others' wrongly classified objects, a strong model can be built. The classifiers mentioned here could be decision trees or logistic regression. The contribution of each model to the ensemble prediction is weighted based on the performance of the model on the training dataset [40].

  [156] used AdaBoost algorithm as machine learning classifier in order to filter Chinese SMS spam messages. [35] used a new and improved AdaBoost approach for sentiment analysis with different base learners for Bagging of airline customer opinions.

- **Bootstrapped Aggregation (Bagging)** A bagging classifier is an ensemble meta-estimator that fits base classifiers each on random samplings of the original dataset and then aggregate their individual predictions (either by voting or by averaging) to form a final prediction. In bagging, a random sample of data in a training set is selected with replacement so that individual data points can be chosen more than once. After several data samples are generated, these weak models are then trained independently where the majority of those predictions is used in the final classification [25].

  [157] a micro-blog emotion classification method is proposed based on a personality and bagging algorithm. [84] proposes a method based on text dimension reduction by manifold learning and Bagging for text classification.

- **Gradient Boost** Gradient boosting classifiers are a group of machine learning algorithms that combine many weak learning models together to create a strong predictive model. Decision trees are usually used when doing gradient boosting [42]. The objective of gradient boosting classifiers is to minimize the loss function, or the difference between the actual class value of the training example and the predicted class value.

  [9] evaluates several machine-learning methods (including gradient boosting) for the sentiment polarity classification of Greek news article user comments. Gradient boosting machines, an ensemble algorithm that can learn with different loss functions. [133] uses gradient boosting algorithm to predict the criminology and the reasons behind the occurrences of the crime. [83] research uses a random-forest classifier to identify the most significant features in high rated apps on Google Play Store. This research uses the gradient boost algorithm to identify the most influential attributes in high rating apps on Google Play Store. To classify the high rated apps, writers use the Gradient Boost algorithm that performs better than Random Forest, K-NN, and Decision Tree algorithm with a 99.93% accuracy.

- **XGB (eXtreme Gradient Boosting)** XGBoost is a refined and customized version of a gradient boosting decision tree system, for improved performance and speed of classification. It is an ensemble of decision trees algorithm where new trees fix errors of those trees that are already

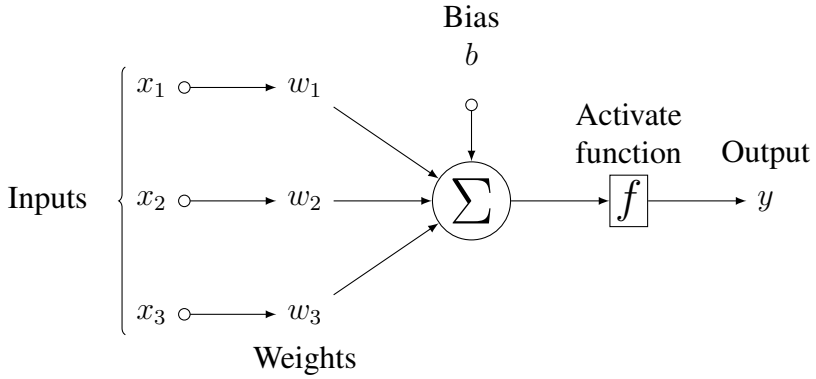


Figure 5: Basic Working Mechanism of Neural Networks.

part of the model. Trees are added until no further improvements can be made to the model [29]. The word "gradient" in gradient boosting refers to the steepness of this loss function, e.g. the amount of error. A small gradient means a small error and, in turn, a small change to the model to correct the error made by the existing state of the ensemble of decision trees. XGBoost is therefore trained to minimize this error.

[130] presented a comparison of the accuracy and speed of execution of the XGBoost algorithm and the gradient boosting algorithm with different datasets terms of accuracy and speed. [85] diabetes prediction based the improved XGBoost algorithm with features combination is 80.2%. [94] presented an improved spam detection model based on Extreme Gradient Boosting (XGBoost) reaching a 2.82% increase in accuracy over an SVM-based spam detector.

2.4.8 Logistic Regression

The logistic regression is a statistical method for predicting binary classes. Logistic regression is named after the function used at the core of the method, the logistic function (or sigmoid function).

[58] uses fast logistic regression with variable length n-grams that can classify different types of texts exploiting the n-gram feature space to automatically provide a compact set of highly discriminative n-gram features. [110] uses POS to approach a multi-class classification problem of authorship attribution for Amazon product reviews with logistic regression. [27] developed a logistic regression spam email filter [99] predicts fake news based on sentiment bias, page rank, and ratio of content length to content errors as fake news discriminants. [87] applied as a multinomial logistic regression form to solve multi-class classification problems.

2.4.9 Deep Learning Algorithms

Deep learning can be considered as a subset of machine learning based on artificial neural networks, which are designed to imitate how humans think and learn. Neural networks are composed of layers of nodes made up of neurons. Nodes within individual layers are connected to adjacent layers. The layers include: an input layer, multiple hidden layers, and an output layer. Data is fed as input to the neurons.

Figure : 5 shows the input values x_i and the weights w_i that determine the importance of the inputs in the artificial neural network. Each entry and its weight are multiplied and summed up together with

the bias b .

$$y = f(x_1 * w_1 + x_2 * w_2 + x_3 * w_3 + \ldots + x_n * w_n + b) \tag{14}$$

An activation function is used to make this sum between 0 and 1 or -1 and 1. To introduce non-linearity into the network we apply non-linear activation functions. Some of the variations of activation function include:

- Sigmoid activation function: The sigmoid activation function is usually used for binary classification problem where output is predicted as false (0) if the value is less than 0.5 else predicted true (1). This strategy is the oldest.
- Tanh activation function is also similar to the sigmoid function but it is symmetric over the origin. It is continuous and differentiable at all points. For some applications, Tanh functions are preferred in hidden layers over sigmoid.
- ReLU (Rectified Linear Unit) activation function was introduced to overcome the vanishing gradient problem. It also accelerates the convergence of stochastic gradient descent and activate all the neurons at the same time.
- Softmax activation function is used to determine the probability associated to each possible output.

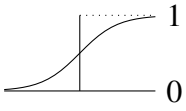
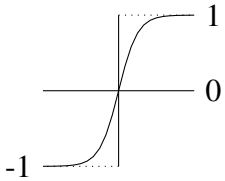
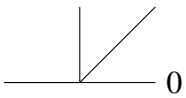
Name	Function	Derivative	Figure
Sigmoid	$f(x) = \frac{1}{1+e^{-x}}$	$f'(x) = f(x)(1 - f(x))^2$	
tanh	$f(x) = \frac{e^x - e^{-x}}{e^z + e^{-z}}$	$f'(x) = 1 - f(x)^2$	
ReLU	$f(x) = \begin{cases} 0 & \text{if } x < 0 \\ x & \text{if } x \geq 0. \end{cases}$	$f'(x) = \begin{cases} 0 & \text{if } x < 0 \\ x & \text{if } x \geq 0. \end{cases}$	
Softmax	$f(x) = \frac{e^x}{\sum_i e^x}$	$f'(x) = \frac{e^x}{\sum_i e^x} - \frac{(e^x)^2}{(\sum_i e^x)^2}$	

Table 2: Non-linear activation functions.

The way in which neurons are organised into layers and connected to each other gives rise to different architectures (or sub-classes) of neural networks such as convolutional and recurrent.

- **Convolutional Neural Network (CNN)** is a multilayer neural network containing two or more hidden layers (see Figure 6). The hidden layers mainly perform two different kinds of functions, namely convolution (extract features (feature map) from the data set) and pooling (subsampling,

is used to reduce the dimensionality of feature maps from the convolution operation). It can automatically extract high-level features from raw input features, which are much more powerful than human-designed features. CNN represents the input data in the form of multidimensional arrays. It works well for a large number of labelled data [2].

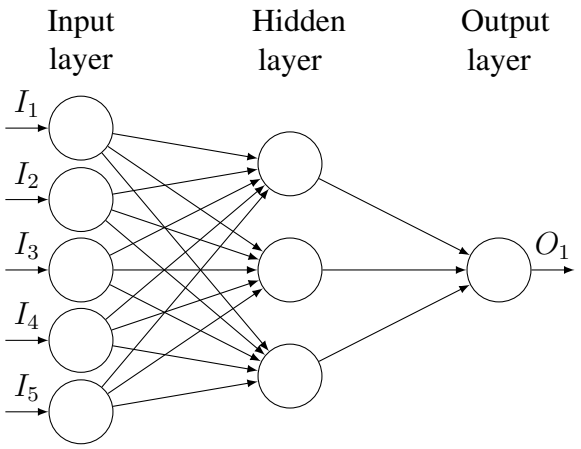


Figure 6: CNN diagram.

[70] proposed a deep convolutional neural network, called FNDNet, which uses GloVe's word-embedding vectors and multiple hidden layers, in order to automatically learn the discriminatory features used to detect the fake news related to U.S. election. [19] developed a neural networks for the Sentiment Analysis (SA) of Twitter text associated with a real application scenario by modifying the network architecture to apply a recurrent pooling layer enabling the learning of longer dependencies between words in tweets.

- **Recurrent Neural Networks (RNNs)** a class of neural networks that allow previous outputs to be used as inputs while having hidden states [143]. The hidden layer saves its output which becomes part of its new input to be used for future prediction. Because of their internal memory, RNN's can remember important signals about the input they received, which allows them to be precise when applied in prediction for some applications. For RNN to makes a decision, the information cycles through a loop. It considers the current input and also what it has learned from the inputs it received previously. RNN has two inputs: the present and the recent past which are related to each other [143].

  An example is provided in Figure 7. In this figure, x_0 is taken from the sequence of input and then it outputs h_0 which together with x_1 is the input for the next step. So, the h_0 and x_1 is the input for the next step. Similarly, h_1 from the next is the input with x_2 for the next step and so on. This way, it keeps remembering the context while training.

  The current state defined by $h_t = f(h_{t-1}, x_t)$ Applying tanh activation function changes the state to $h_t = tanh(w_{t-1}h_{t-1} + w_t x_t)$ will give an output of $y_t = f(w_{ty}x_t)$

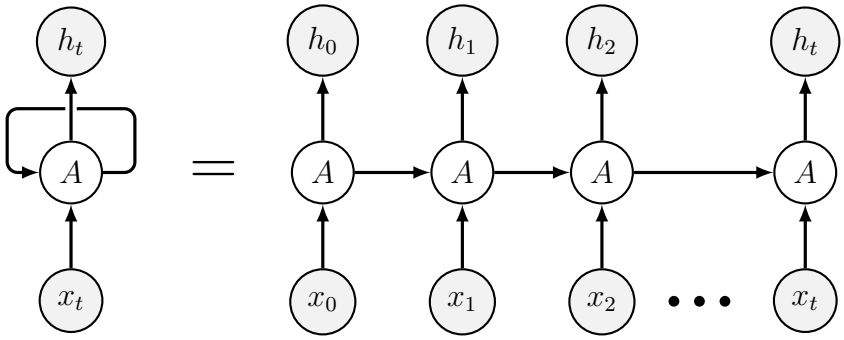


Figure 7: RNN diagram.

[44] use recurrent neural networks to analyse the reviews to automatically predict the user ratings based on the reviews by applying transfer learning from a huge volume, gold dataset of Amazon customer reviews. [82] used the RNN architecture with Word2vec for sentiment analysis in Indonesian. [150] used recurrent neural networks (RNN) for the classification of unwanted and normal messages and obtained an accuracy of 98%. [64] presented a framework that spots and classifies fake news messages using improved Recurrent Neural Networks and Deep Structured Semantic Model.

- **Long Short-Term Memory Networks (LSTMs)** is a special kind of RNN with a feedback connections, capable of learning long-term dependencies allowing information to persist as well as selectively remembering patterns for long duration of time [49]. A common LSTM unit is composed of a cell, an input gate, an output gate and a forget gate. The cell remembers values over arbitrary time intervals and the three gates regulate the flow of information into and out of the cell [49].

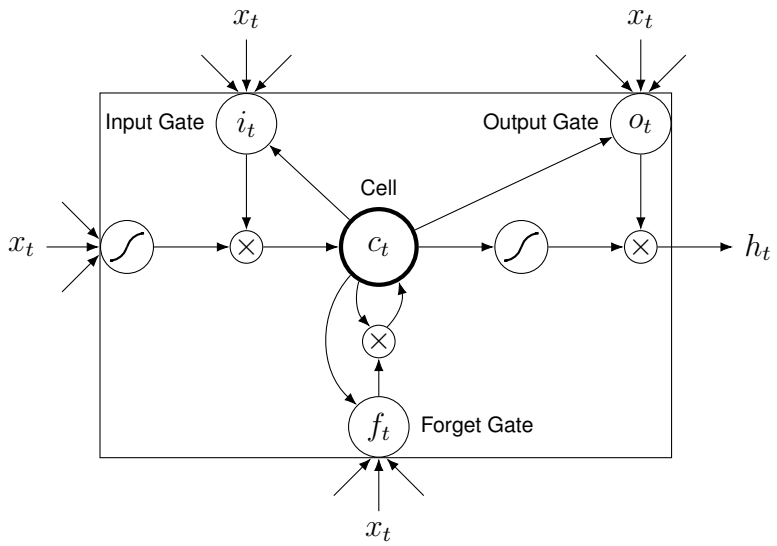


Figure 8: LSTM diagram.

The three gates present in LSTM are:

  - Input gate discover which value from input should be used to modify the memory. Sigmoid function decides which values to let in the range [0; 1]. While tanh function gives weights to the values which are passed deciding their level of importance ranging from -1 to 1. In Figure 8, one can see that the input gate is shown with the related equation $i_t = \sigma(w_i * [h_{t-1}, x_t] + b_i)$ while the centre cell $C_t = tanh(w_c * [h_{t-1}, x_t] + b_c)$

– Forget gate discover what details to be discarded from previous seen data. It is decided by the sigmoid function. It looks at the previous state h_{t-1} and the content input x_t and outputs a number between 0 (omit this) and 1 (keep this) for each number in the cell state C_{t-1}. $f_t = \sigma(w_f * [h_{t-1}, x_t] + b_f)$

– Output gate generates the output o_t and h_t . Sigmoid function decides which values to let through [0; 1]. The tanh function gives weights to the values which are passed deciding their level of importance ranging from -1 to 1 and multiplied with output of Sigmoid. $o_t = \sigma(w_0 * [h_{t-1}, x_t] + b_0)$ $h_t = o_t * tanh(C_t)$

[10] proposed a bi-directional long short-term memory (LSTM) model to detect misinformation. [153] presents a complete solution of LSTM network hardware implementation based on memristor crossbar for sentimental analysis.

- **Gated Recurrent Unit Networks (GRU)** uses the same workflow as an RNN, but each GRU unit's operation and associated gates are different. GRU uses the update gate and reset gate gate operating techniques to address the issue with traditional RNN.

  The amount of prior knowledge that must be transmitted along with the next state is decided by the update gate. This is highly effective since the model has the option of copying all historical data and removing the possibility of vanishing gradients.

  The reset gate is utilized in the model to determine how much of the prior knowledge must be disregarded; in other words, it determines whether or not the previous cell state is significant.

  When the reset gate first enters operation, it stores pertinent data from the previous time step in new memory content. The input vector, hidden state, and their weights are then multiplied. It then computes the element-wise multiplication between the reset gate and the previously hidden state multiple. The following sequence is formed by applying the non-linear activation function following the summarization of the preceding steps.

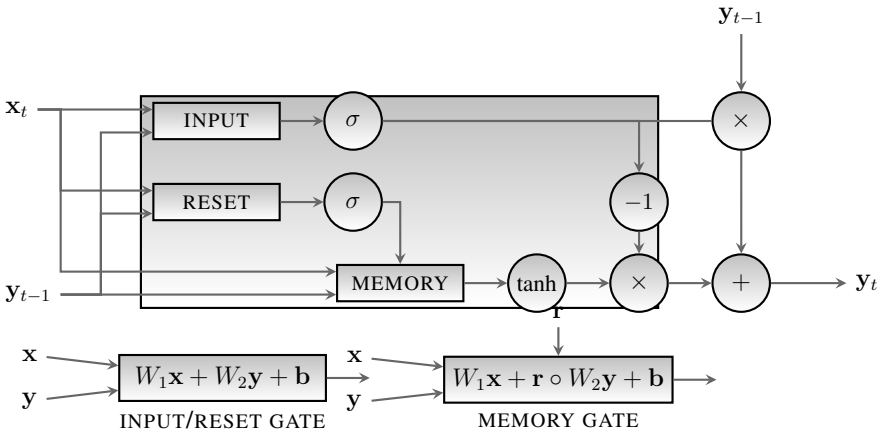


Figure 9: GRU diagram.

- **Echo state networks (ESN)** type of (recurrent) network with random connections between the neurons (i.e. not organised into neat sets of layers). Figure 10 displays the general architecture of this kind of NN. It is a recurrent neural network with a loosely connected hidden layer, called a 'reservoir'. To train this network we feed the input, forward it and update the neurons for a while, and observe the output over time. During training, only the connections between the observer and the hidden units are changed [86]. Only weights in the output neuron are trained in order to reproduce specific temporal patterns. Weights in the input and hidden (reservoir) layers are randomly assigned and not trainable [86].

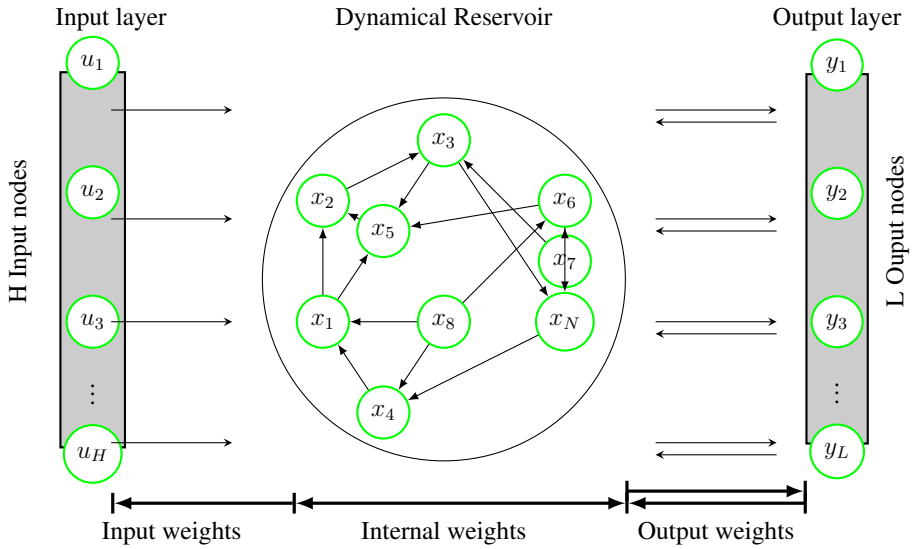


Figure 10: ESN diagram.

Calculating the state of the hidden neuron in the dynamic reservoir we get $X_{n+1} = f(W_{in} * U_n + W_{dr} * X_n + W_{bk} * D_n)$ where f is the activation function of the hidden neurons, W_{in} input weight, W_{dr} is the hidden weight W_{bk} it the feedback weight D_n is the teacher for the training mode. The output neuron is calculated by: $Y_{n+1} = f_{out}(W_{out}[u_{n+1}, x_{n+1}, y_n])$. f_{out} is the activation function for the output neuron.

[114] explored a random contextual-word encoder using echo state networks for named entity recognition. [134] developed a character trigrams to investigate the performance of Echo State Network-based Reservoir Computing (ESN) on a text document classification task with 15 authors of the C50 Reuters dataset.

2.5 Evaluation of Models

Model evaluation aims to estimate the generalization accuracy of a model on future (unseen) data. Model evaluation metrics are required to quantify model performance. The choice of evaluation metrics depends on a given machine learning task. A confusion matrix provides a more detailed breakdown of correct and incorrect classifications for each class. Some of the metrics used in classification problem are:

		Actual values		
		Positive	Negative	Total
Predicted values	Positive	$True\ Positive(TP)$	$False\ Positive(FP)$	$TP + FP$
	Negative	$False\ Negative(FN)$	$True\ Negative(TN)$	$FN + TN$
	Total	$TP + FN$	$FP + TN$	n

Table 3: Confusion matrix

- **True positives** are when an observation is predicted to belongs to a class and it actually does belong to that class.
- **True negatives** are when an observation is predicted not belong to a class and it actually does not belong to that class.

- **False positives** occur when you predict an observation belongs to a class when in reality it does not.
- **False negatives** occur when you predict an observation does not belong to a class when in fact it does.

Based on the contingency table presented in table 3, one can define the following performance measures.

Accuracy is defined as the percentage of correct predictions for the test data. It can be calculated easily by dividing the number of correct predictions by the number of total predictions.

$$Accuracy = \frac{correct\ predictions}{all\ predictions}$$

Precision is defined as the fraction true positives among all of the examples which were predicted to belong in a certain class.

$$Precision = \frac{True\ Positives}{True\ positives + False\ Positives}$$

Recall is defined as the fraction of examples which were predicted to belong to a class with respect to all of the examples that truly belong in the class.

$$Recall = \frac{True\ Positives}{True\ Positives + False\ Negatives}$$

F1-score combines precision and recall into a unique value [107].

$$F_1 = 2 * \frac{Precision * Recall}{Precissions + Recall} = \frac{TP}{TP + 1/2(TP + FN)}$$

Area under the curve (AUC) measures the ability of systems to assign higher scores to positive cases in comparison to negative cases [107].

c@1 measures the accuracy of binary predictions but also the ability of systems to leave difficult cases unanswered [108].

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

Where n, n_c and n_u denote the number of problems, the number of correct answers and the number of unanswered problems, respectively.

F_0.5u a measure that puts more emphasis on deciding same-author cases correctly and rewards non-answers[16].

Brier is a score used for evaluating the goodness of (binary) probabilistic classifiers. It is the complement of the Brier score loss function [24].

3 Feature Selection

There are numerous features to consider when developing a machine learning model for text classification. Because the features are composed of words, sequence of letters or words, the number of possible features can be huge (e.g. in K or G) Of course one can use all of them and let the learning scheme select the "best" subset. However to get the desired result, these characteristics necessitate a significant amount of time and computational power. For decades, feature selection has been a research area in different fields, such as bioinformatics, image recognition, image retrieval and text mining.

3.1 The Curse of Dimensionality

The curse of dimensionality is a phenomena that describes how problems are extremely difficult to solve due to the increasing number of features. The search for the best feature subset frequently surpass available computing capabilities. Another point of concern is closely related to high-dimensional data. Because data in highly dimensional vector spaces is extremely sparse, estimating any parameter requires a large number of samples to obtain a respectable degree of accuracy. Moreover the sample must cover the entire feature space and not only a small fraction. With the amount of features, the dimensionality of feature space increases exponentially. Furthermore, because the resulting computer power requirement is excessively large, this drastically limits conceivable applications and substantially limits a potential set of solutions. Of course, this is true in the field of document classification as well.

For example, in a blog corpus [137], the number of words occurring in more than five posts written by men is 97,089. When all conceivable bigrams of words are considered, the number of characteristics could reach the value of 9,426M (=97,089 x 97,089). Even if just half of them occur, the sample size makes any analysis difficult. Furthermore, the amount of time it takes to get an attribution is proportional to the size of the feature set. For example, in CLEF-PAN 2014 [118], determining the gender of an author could take more than 3 hours (for 154 documents), but other methods take less than 1 minute. As another example, one system required more than 37 hours to provide a solution for the 25 documents in the cross-domain authorship attribution task in 2019 [74] .

The more variables you have, the more samples you'll need to represent all the different combinations of feature values in the example. As the number of variables in the model grows, the model becomes more complex, raising the risk of overfitting. When you train an ML model on a huge dataset with a lot of features, it's inevitable that it'll be reliant on the training data. As a result, the model will be overfitted and will not perform well on real data. In fact, some random association between features and the decision are viewed as significant by the learning scheme. Avoiding overfitting is the fundamental goal of dimensionality reduction. Training data with considerably lesser features will ensure that your model remains simple and easy to interpret for a human being.

Other advantages of dimensionality reduction include, first the elimination of noise and superfluous features. Second, it aids in the accuracy and performance of the model. Third, it compresses the data, reducing computing time and allowing for faster data training.

3.1.1 Lasso Regression

When developing a model, not all of the features in our training data are equally essential. If we had enough computational resources, we could include all of the available features in our model, but this has (at least) two drawbacks. It can lead to overfitting and reduces the interpretability of our model.

Building a sparse model, which is based on a subset of the most relevant features, is significantly more useful. The Lasso (least absolute shrinkage and selection operator) regression is a model capable of providing its own interpretation of feature importance [41].

By combining variable selection and regularization, Lasso regression improves the predictability and interpretability of the final model. As discussed below, it gives a rational technique to decrease the amount of features in a model.

Regularization, often known as 'shrinkage' is the process of reducing the size of coefficients associated with the features. In linear regression, it is common to minimize a cost function (a loss function or objective function) known as the residual sum of squares (RSS) when determining the best fit.

$$RSS(\boldsymbol{\beta}) = \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 \tag{15}$$

where β_j are the regression coefficients associated with each feature and $\boldsymbol{\beta}$ is a vector of all β_j values.

By including a penalty term (or regularization term) in our cost function, we can reduce the regression coefficients a process also known as 'shrinkage' or ridge regression.

$$PRSS = RSS(\boldsymbol{\beta}) + \lambda \sum_{j=1}^{p} \beta_j^2 \tag{16}$$

where the 'tuning parameter' is $\lambda > 0$. We resort to the previous residual sum of squares cost function if $\lambda = 0$. With $\lambda > 0$, the cost function has a greater value when the coefficients associated to the features are higher. The goal of Lasso regression is to reduce the absolute values of the coefficients while optimizing the cost function. Cross-validation is used to find the best tuning parameter value λ. At the end, features with small β_j values can be removed to obtain a better and simpler model

We now have the following penalty term with Lasso (Least Absolute Shrinkage and Selection Operator) regression.

$$PRSS = RSS(\boldsymbol{\beta}) + \lambda \sum_{j=1}^{p} |\beta_j| \tag{17}$$

The l_1 norm, also known as the Manhattan distance, has the effect of driving some of the coefficients to zero, effectively deleting features from the model and performing feature selection.

In order to minimize the cost function, Lasso regression will automatically identify the features that are advantageous while discarding those that are useless or redundant. A feature's coefficient becomes 0 (or close to 0) when it is eliminated from a Lasso regression. So, Lasso regression is conducted on a scaled version of the dataset and only evaluate features with coefficients greater than 0. Hyperparameter tuning is done before performing the correct type of Lasso regression.

3.2 Two-stage feature selection

We use an idea from the l_1 norm of Lasso regression, which was used to reduce redundant features, to be able to focus on the important stylistic markers capable of discriminating across groups of writers. In our situation, the distance between the occurrence probability in each class is taken into account. We utilize both the positive and negative distance to determine which category a feature in a tweet or document belongs in. In the case of binary classification, a positive distance indicates that the term (feature) belongs to one class and a negative distance indicates that it belongs to the other.

In our work, we used this concept to provide a two-stage feature selection technique to overcome some of the word frequency difficulties. Term frequency (TF) information is useless in an author profiling task and the difference between attributes associated with each author group is not assured.

The TF information is taken into account in the first case. The discriminative power of each term is calculated by estimating the difference in the likelihood of occurrence in both categories, as shown in Equation (18). In our example, assuming we have two categories, namely men (M) or women (F) the absolute frequency of the i_{th} term in a tweet in category M (male) is indicated by tf_{iM}, while the text length (in tokens) of all tweets in class c_M is indicated by n_M [61].

$$probD(t_i, c_M) = prob(t_i, c_M) - prob(t_i, c_F) = \frac{tf_{iM}}{n_M} - \frac{tf_{iF}}{n_F} \tag{18}$$

Only terms with a positive probD value are retrieved to find terms that can characterize the male group. To represent the feminine group, only words with a negative probD value are chosen. This process creates two term clusters, one for each class.

When faced with more than two classes, one can apply the following equation.

$$probD(t_i, c_j) = prob(t_i, c_j) - \sum_{k=1, k\neq j}^{r} prob(t_i, c_k) \tag{19}$$

As shown in Equation (19), this approach can be generalized to more than two categories. The discriminative strength of term t_i for the j^{th} category in this situation is determined by the occurrence probability over all $r - 1$ categories.

In a separate phase, infrequently occurring terms could be ignored. Terms with a document frequency (DF) value of less than or equal to three are deleted to achieve this.

The goal of this culling technique is to eliminate terms that have a low overall occurrence frequency or a high term frequency but localized to a few documents (unbalanced terms). For example, with the male category CLEF-PAN-15ET corpus, (the word "francisco", for example, only appears in two documents with a TF of 51 (as does the word "castelletti ", which has an occurrence frequency of 43 but only appears in two documents). Even with a high TF, those terms are ignored due to a small DF value.

With this feature reduction, with the male category (CLEF-PAN-15ES corpus), the number of features is reduced from 14,276 (vocabulary size used by men) to 930 (a decrease of 93.5%). For the women class, the number of terms decreases from 13,990 to 955 (93.3%). Similar feature reductions are shown for other languages in the same corpus in Table 4.

The proposed solution is relatively straightforward to implement when compared to existing feature selection strategies based on a filter-based ([138] [158]) or wrapper approach ([38] [154]) approach. The basic idea is to integrate the word and document frequency components to find often occurring terms with a strong relationship to each target category. Furthermore, each category can be identified by m attributes. A Chi-square-based filtering strategy or a wrapper solution don't have this feature. With these two feature selection strategies, a category could be overlooked or have significantly less features than the others. The proposed technique avoids this issue.

In comparison to PMI (Pointwise mutual information) and χ^2 (Chi-square) feature ranking, our method may exclude uninformative terms, leaving only the relevant set for further investigation. The terms are prioritized according to their importance in the PMI and χ^2 techniques, and it is up to the user to select how many terms are required.

Language		All features	tf > 1	df > 3	decrease (%)
2015_EN	Male	13299	4852	1113	91.63
	Female	12689	4689	1051	91.72
	Total	25988	9541	2164	91.67
2015_ES	Male	14276	4918	930	93.49
	Female	13990	4629	955	93.31
	Total	28266	9547	1885	93.33
2015_IT	Male	7092	2080	355	94.99
	Female	7699	2317	395	95.39
	Total	14791	4397	750	95.39
2015_NL	Male	5011	1615	352	92.98
	Female	5271	1761	312	94.08
	Total	10282	3376	664	93.54

Table 4: Percentage of decrease in number of feature.

In each of the term selection mentioned, we present some of the terms selected by the methods in Table 5 and 6. We observe that the terms set selected by each method is different. PMI and our method assign terms to each target category but this is not the case for χ^2.

two-step feature selection

features	+probD	features	-probD
url	0.0063259607	*	-0.0000003140
the	0.0028386779	movie	-0.0000007479
at	0.0018389699	wonder	-0.0000011819
of	0.0010695191	art	-0.0000011819
learn	0.0006799185	kill	-0.0000016158
other	0.0006768069	first	-0.0000022438
game	0.0006435137	hand	-0.0000024837
for	0.0006110953	few	-0.0000033515
post	0.0005971284	super	-0.0000046534
use	0.0004976546	nobody	-0.0000046534
analysis	0.0004961129	together	-0.0000046534
be	0.0004888306	catch	-0.0000046534
and	0.0004878467	complete	-0.0000046534
google	0.0004550547	how	-0.0000049956
big	0.0004409747	water	-0.0000050873
no	0.0004402726	date	-0.0000050873
language	0.0004347797	color	-0.0000055212
just	0.0004273462	notice	-0.0000059552
process	0.0004098513	ride	-0.0000063891
air	0.0004025943	celebrate	-0.0000063891
any	0.0003966391	hide	-0.0000063891
look	0.0003846547	challenge	-0.0000063891
fuck	0.0003635118	introduce	-0.0000063891
badge	0.0003390914	attack	-0.0000063891
play	0.0003216598	sick	-0.0000068230

Chi2 feature selection

features	Chi2 value
!	174.3863811
urllink	146.6329677
data	82.59159216
love	68.12642544
courtesy	64.37954846
startup	53.39186374
you	52.08581871
plastic	49.80129868
francisco	44.48234817
pic	44.40336567
castelletti	44.32865934
my	39.65989352
????	37.64825241
education	37.5845506
analysis	36.92500767
nowplaying	35.33174704
at	34.56106621
sentiment	34.1693187
happyday	33.99559771
nlproc	31.93073985
weekly	31.89064761
stem	31.82643494
skill	30.85873527
google	30.28473512
process	30.25592543

Table 5: Some of the selected features with our FS method and Chi2 .

PMI feature selection

features	male PMI value	features	female PMI value
castelletti	1.019890544	plastic	0.9803738694
andrew	1.019889778	chelseafringelj	0.9803718893
nlproc	1.019889246	ary	0.9803709275
bueno	1.019889091	songsonshuffle	0.9803701405
córdoba	1.019888925	photoedit	0.9803698283
wcpr	1.019887627	abundance	0.9803694848
interesante	1.019886306	portrait	0.9803691051
recruitment	1.019885885	chattanoogastrong	0.9803686833
afterlight	1.019885413	modena	0.9803682118
jorge	1.019883595	ljubljana	0.9803663933
uno	1.019882802	stress	0.9803656006
mufc	1.019881877	pill	0.9803646758
buster	1.01987787	tedxbrussel	0.9803635829
daedalus	1.01987329	hgh	0.9803606684
palestra	1.019869855	modernvintage	0.9803586647
hiddengemsapp	1.019865046	swine	0.9803560885
teamgermany	1.019857833	foreal	0.9803526535
politecnico	1.019845811	athena	0.9803406313
vedere	1.019821767	flush	0.9803045653
aphrodite	1.019749637	fatigue	0.9802324359
francisco	0.9888670039	photography	0.897914538
sentiment	0.9803655354	bird	0.892913857
courtesy	0.9780737239	cancer	0.8808410247
semantic	0.9709842991	weekly	0.8784970842
iemand	0.9557635621	wrap	0.8648994808

Table 6: Some of the selected features using PMI.

4 Author Profiling

Author profiling (AP) is the task of extracting demographic aspects of the author of a text. The information can be psychological (i.e. author personality, mental health, native language speaker/not), sociological (e.g. age range, gender, education level, origin) or other information related to the author.

We present in this section the state of the art and the different applications of author profiling providing examples of recent works done in this field.

4.1 Introduction

Author profiling is the analysis of a given set of texts in an attempt to uncover various characteristics that distinguishes between classes of authors to be able to determine an author's gender, age range, native language, personality type [125]. Author profiling techniques are of growing importance due to the huge amount of texts that are available on social media platforms. Automatically predicting the identity of authors has a number of potential applications such as forensics, security, and marketing.

In marketing, knowing the gender, age range and personality of the customer will allow companies to direct their advertisements to the targeted customers. To know who liked and disliked their products, companies may analyse on-line reviews to improve targeted advertising in order to achieve a better market segmentation [149].

Identifying the author of the threatening text is the first step in countering it. For forensics application, the characteristics of the author of an anonymous text or harassing text messages can be analysed for criminal investigation. In security aspect knowing the background, demographics and native language of an author can lead to potential identification of terrorists [31]. For fake news detection, understanding the different linguistic patterns of social media uses will enable these platforms to effectively examine the user content and flag off and inform users about which pieces of news contain fake information [6].

Every human has a distinctive writing style related to his/her education, origin, gender. This background is carried on to different platforms such as twitter, social media, reviews, blogs and documents. By exploiting the textual content of these platforms, the demographic features of the authors can be determined [47].

4.2 State of the art

The following sections describe the related work for age range, gender, language variety identification as well as fake news detection.

Age range and gender identification NLP-based tools based on stylometric models have been applied to profiling users by determining their attributes like age range and gender by examining the use of language [8]. For example, the variation in pronouns frequency can be used to identify the gender. In this case women use the first person of singular more than men and the way men use more determiners. Author profiling pioneer researchers focused mainly on texts such as books, newspapers, magazines, and blogs. [7] achieved approximately 80% classification accuracy by combining function words with parts-of-speech (POS) tags. Usually corpora of written texts can be extracted from the British National Corpus.

Currently researchers focused mainly on social media such as Twitter and Facebook posts, blogs, customer reviews where the language is more spontaneous and less formal. [135] studied the effect of age range and gender in the style of writing in blogs. They gathered over 71,000 blogs and obtained a set of stylistic features like non-dictionary words, parts-of-speech, function words and hyperlinks, combined with content features, such as word. They obtained an accuracy of about 80% for gender identification and about 75% for age range identification. [79] achieved approximately 80% accuracy on the task of automatic determination an author's gender by proposing combinations of simple lexical and syntactic features. [48] added some new features as slang words and the average length of sentences, improving accuracy to 80.3% in age range group detection and to 89.2% in gender detection. [96] studied the use of language and age range among Dutch Twitter users from blogs, telephone conversations transcripts, and online forum posts and found both aesthetic features (such as POS patterns) and content-oriented features are effective when used with linear regression model.

The automatic detection of bots is another author profiling duty. Bots are computer programs that imitate humans in order to influence users for commercial, political, or ideological reasons. Due to their goal of spreading disinformation and hate speech, malicious bots are strongly linked to polarization. [13] noted the use of bots to enhance some political opinions or supporting some political candidates during elections discovering that in the week before election day, around 19 million bots tweeted to support Trump or Clinton. [146] demonstrated that bots were responsible for 23.5 percent of the 3.6 million tweets about the Catalan independence referendum on October 1, 2017. Both parties received emotional and angry remarks from these bots. Bots could artificially raise a product's popularity by promoting it and/or posting positive reviews, as well as smear the reputation of competing products by assigning poor ratings. When the goal is political or health the danger is significantly greater. Hence the need to investigate whether the author of a Twitter feed is a bot or a human a task that was performed in CLEF-PAN 2019 under the title "Bots and Gender Profiling in Twitter" [18].

Another challenge in author profiling is language variety identification. Information on how people exchange language can be used to differentiate between classes of writers based on their language variety, a task that is especially crucial in social media. In Arabic, [127] reported n-gram accuracy of 98% in six Arabic dialects: Egyptian, Iraqi, Gulf, Maghreb, Levantine, and Sudan. With a combination of content and style-based features, [34] showed 85.5% accuracy differentiating between Egyptian and Modern Standard Arabic. In [88], the authors explored the identification of Argentinian, Chilean, Colombian, Mexican, and Spanish in Twitter, reporting accuracies of 60-70% using a combination of n-grams and language models. In

The ease with which people can publish anything on social media has resulted in an increase in the amount of misinformation being published and disseminated, which has had a number of negative implications in society. As a result, the author profiling community is also interested in automatic fake news identification. [124] studied language elements such as personal pronouns and swear words, which were then put into a Long Short Term Memory (LSTM) network to determine believability. Other studies suggested that the emotions portrayed in the news be used. In this vein, [46] suggested emoCred, an LSTM-based neural network that uses emotions from text, while [45] advocated incorporating emotions retrieved from text into an LSTM network and shown that emotions are beneficial for classifying different types of fake news. Guoet al. [51] proposed a dual emotion-based fake news detection framework for publishers and users, respectively, to learn content and comment emotion representations, whereas, Wang [152] presented a hybrid convolutional neural network that combines user metadata with text. To boost the participation of detecting probable fake news spreaders on Twitter as a first step towards preventing fake news from being circulated among social media users, [104] presents Author Profiling Task at CLEF-PAN 2020: Profiling Fake News Spreaders on Twitter.

The task of obtaining author profiles is gaining traction in the scientific community, as seen by the

number of similar projects that have cropped up in the previous years. One can mention the BEA-8 Workshop at NAACL-HT on Native Language Identification [89] as well as the job on Computational Personality Recognition (WCPR) at ICWSM 2013 and at ACM Multimedia 2014 [26] and CLEF-PAN evaluation campaign since 2013 to present [102][106][18][104] .

4.3 Corpora

We evaluate our learning models using a variety of datasets. The evaluations in this thesis are based on datasets from CLEF-PAN evaluation projects. These datasets have been generated to predict gender, age range, and language diversity. We also use the information to predict fake news spreaders. The datasets are described in the same way as they were provided for the CLEF-PAN shared tasks, respecting the splitting into the training and testing samples.

4.3.1 PAN 2014, Age range and Gender

During the 2014 year, the corpus contains four different genres: social media, blogs, Twitter, and hotel reviews provided in English and Spanish. The hotel reviews were only available in English. The corpus documents are encoded as XML files, one per author, with the contents between two <document> tags. The author is labelled with age range and gender information. Four age range classes are defined a) 18-24; b) 25-34; c) 35-49; d) 50-64; e) 65+ [122]. The gender classifications are balanced. However the age range categorization task is extremely imbalanced. The 65+ age range, in particular, has extremely few examples. Table 7 provides an overview of the language and age range classes.

The social media corpus contained those authors whose posts contained an average of 100 posts per author. During the evaluation of the corpus, fake profiles were also removed such as authors selling the same product and authors with a high number of text reuse.

	Training		Testing	
	English	Spanish	English	Spanish
Female	3873	636	1688	283
Male	3873	636	1688	283
18 -24	1550	330	680	150
25 - 34	2098	426	900	180
35 - 49	2246	324	980	138
50 - 64	1838	160	790	70
65+	14	32	26	28
$\sum$	7746	1272	3376	566

Table 7: CLEF-PAN 2014: Distribution of social media with respect to age range classes per language balanced by gender

Blogs were created with the help of linkedIn website. The existence of the blogs were verified to determine the language of interest, if they are updated by only one person, the age range and gender of the author. Blogs are discarded if it is not updated by a single author. In addition, the blogs were ignored if either person, education level, age range and gender are not clear. Table 8 represents an overview of the blog sub corpus with the distribution in language and author age range shown.

In total, 131 Twitter profiles from several domains (energy, environmental, banking, automotive, and corporate social responsibility sectors) were annotated with age range and gender. As one can see in

	Training		Test	
	English	Spanish	English	Spanish
Female	73	44	39	28
Male	74	44	39	28
18 -24	6	4	10	4
25 - 34	60	26	24	12
35 - 49	54	42	32	26
50 - 64	23	12	10	10
65+	4	4	2	2
$\sum$	147	88	78	56

Table 8: CLEF-PAN 2014: Distribution of blogs with respect to age range classes per language

Table 9, the list of profiles were not balanced by age range because influential Twitter authors in the considered economic domains tend to be male and of quite a narrow age range (35-49).

	Training		Test	
	English	Spanish	English	Spanish
Female	153	89	77	45
Male	153	89	77	45
18 -24	20	12	12	4
25 - 34	88	42	56	26
35 - 49	130	86	58	46
50 - 64	60	32	26	26
65+	8	6 2	2	2
$\sum$	306	178	154	90

Table 9: CLEF-PAN 2014: Distribution of Twitter with respect to age range classes per language

Hotel review corpus was crawled from the hotel review site TripAdvisor in the period of one month from mid February to mid March 2009. This corpus contains 235 793 reviews about 1,850 different hotels. Each review comprises its author's user name, the review text, and the date the review was written. After post-processing steps (removing short reviews, reviews without age range and gender information, text was not written in English). The corpus was reduced to 58,101 reviews and covers six age range classes. To obtain a nearly uniform age range class distribution, 700 authors were sampled from each of the three major classes (25–34, 35–49, 50–64). Table 10 exposes the distribution of gender and age range classes. The class 13–17 was discarded completely since the number of available authors was found to be not representative for evaluation purposes.

4.3.2 CLEF-PAN 2015: Age range and Gender

The corpus extracted from Twitter in 2015 is available in the four languages namely English, Spanish, Italian, and Dutch. It is annotated with gender and personality traits as well as with age range classes (English and Spanish only). The age range and gender information was reported by the Twitter users themselves which were labelled according to the following classes: a) 18-24, b) 25-34, c) 35-49, and d) 50+. Table 11 shows the distribution of age range and gender classes for the given languages. Personality traits were self-assessed with the BFI-10 on-line test and reported as scores normalized

		Webis-TripAd-13		Training set		Test set	
Gender	Age range	Authors	Reviews	Authors	Reviews	Authors	Reviews
female	13 -17	23	23	-	-	-	-
	18 -24	656	741	180	208	74	84
	25 - 34	7517	9504	500	651	200	247
	35 - 49	10554	13552	500	659	200	255
	50 - 64	5850	7449	500	617	200	242
	65+	547	682	400	494	147	188
male	13 -17	22	25	-	-	-	-
	18 -24	254	1314	180	228	74	86
	25 - 34	3816	5144	500	700	200	250
	35 - 49	8586	12044	500	707	200	302
	50 - 64	5413	7229	500	667	200	268
	65+	1079	1394	400	520	147	178

Table 10: CLEF-PAN 2014: Distribution of reviews with respect to age range classes per language

between -0.5 and +0.5. The corpus is balanced with respect to gender, but the skew of the age range distribution is considerable due to the lower number of aged 50 and older using Twitter [100].

	Training				Test			
	EN	ES	IT	DU	EN	ES	IT	DU
Female	76	55	19	17	71	44	18	16
Male	76	55	19	17	71	44	18	16
18 -24	58	22			56	18		
25 - 34	60	56			58	44		
35 - 49	22	22			20	18		
50 +	12	10			8	8		
$\sum$	152	110	38	34	142	88	36	32

Table 11: CLEF-PAN 2015: Distribution of Twitter users with respect to the labels in the corpus per language

4.3.3 CLEF-PAN 2016: Cross-genre Age range and Gender Identification

For 2016, CLEF-PAN organizers have generated a corpus written in English, Spanish and Dutch. The authors are labelled with age range and gender information, except in case of Dutch where only gender information is provided Table 12 exposes the distribution of age range and language classes. For labelling age range, the following classes were considered: a) 18-24; b) 25-34; c) 35-49; d) 50-64; e) 65+. The training part was collected for the three languages. Test corpora in the Dutch sub-corpus were collected from reviews, whereas in English and Spanish the test corpus was collected from social media, and the test corpus was collected from blogs [103]. The dataset is divided into training/test in a 60/40 proportion, with 300 authors for training and 200 authors for test.

	Training (Twitter)		Test (Blogs)	
Age range	English	Spanish	English	Spanish
18 -24	26	16	10	4
25 - 34	136	64	24	12
35 - 49	182	126	32	26
50 - 64	78	38	10	10
65+	6	6	2	4
$\sum$	428	250	78	56

Table 12: CLEF-PAN 2016: Distribution of authors with respect to age range classes per language (English and Spanish)

4.3.4 CLEF-PAN 2017: Gender and Language Variety Identification

For 2017 the CLEF-PAN focus was on language variety identification. To achieve this, Twitter was selected as the corpus source with English, Spanish, Arabic and Portuguese languages. For each variety, the capital (or more populated cities) of the region where this variety is used is selected. For each selected author, the tweet contains at least 100 tweets, ignoring all re-tweets. An author is annotated with a corresponding language variety when it has been retrieved in the corresponding region moreover, at least 80% of the locations provided as meta-data of her/his tweets must coincide with some of the toponyms for the corresponding region. The final dataset is balanced in the number of tweets per variety and gender, and in the number of tweets per author: 500 tweets per gender and variety 100 tweets per author [102]. Table 13 provides an overview of this corpus.

Language	Variety	City
Arabic	Egypt	Cairo
	Gulf	Abu Dhabi, Doha, Kuwait, Manama, Mascate, Riyadh, Sana'a
	Levantine	Amman, Beirut, Damascus, Jerusalem
	Maghrebi	Algiers, Rabat, Tripoli, Tunis
English	Australia	Canberra, Sydney
	Canada	Toronto, Vancouver
	Great Britain	London, Edinburgh, Cardiff
	Ireland	Dublin
	New Zealand	Wellington
	United States	Washington
Portuguese	Brazil	Brasilia
	Portugal	Lisbon
Spanish	Argentin	Buenos Aires
	Chile	Santiago
	Colombia	Bogota
	Mexico	Mexico
	Peru	Lima
	Spain	Madrid
	Venezuela	Caracas

Table 13: CLEF-PAN 2017: Cities selected as representative of the language varieties

4.3.5 CLEF-PAN 2018: Gender Identification

In 2018 CLEF-PAN organisers have decided to create a collection reflecting two media, namely texts and images. Therefore the corpus is made of textual and image information framed as a multilingual

task, covering the languages Arabic, English, and Spanish. The corpus is balanced regarding gender and contains 100 tweets and 10 images per author [106] (see Table 14) .

	(AR) Arabic	(EN) English	(ES) Spanish	Total
Training	1,500	3,000	3,000	7,500
Test	1,000	1,900	2,200	5,100
Total	2,500	4,900	5,200	12,600

Table 14: CLEF-PAN 2018: Number of authors per language and subset.

4.3.6 CLEF-PAN 2019: Bot or a Human, in case of human, Gender of the author.

For 2019, the CLEF-PAN task is focusing on detection of bots on tweets. Thus the main task was to identify set of 100 tweets written by humans or bots. Bot twitter counts were identified and manually annotated with agreement of at least two annotators. The corpus is completely balanced per type (bot / human), and in case of human, it is also completely balanced per gender. Each author is composed of exactly 100 tweets [101]. The distribution over the two languages, bots vs human and and author gender is displayed in Table 15.

4.3.7 CLEF-PAN 2020: Profiling Fake News Spreaders on Twitter

The corpus was created by reviewing fact-checking websites such as PolitiFact or Snopes to find news labelled as fake. These news containing fake news were searched on Twitter and manually inspected them to discard those not actually referring to the news. The corpus consists of 500 authors for each of the two languages, English and Spanish. The corpus for each language is balanced,with 250 authors for each class (fake and real news spreaders). The corpus was split into training and test sets, following the 60/40 proportion [104]. Table 16 contains the distribution within this corpus.

4.3.8 CLEF-PAN 2021: Profiling Hate Speech Spreaders on Twitter

The CLEF-PAN 2021 corpus was built by searching for people who could be deemed prospective haters. To accomplish so, two ways were used: (i) a keyword-based strategy (e.g., looking for angry passages aimed mostly at women or immigrants); and (ii) a user-based approach (e.g., inspecting persons identified as haters and tracking their networks) (e.g. followers and followees). Second, the timelines of the specified users were collected, and those tweets conveying hate were manually annotated. Finally, users who have more than ten hostile tweets are labelled as "keen to promote hate

	(EN) English				(ES) Spanish			
	Bots	Female	Male	Total	Bots	Female	Male	Total
Training	2,060	1,030	1,030	4,120	1,500	750	750	3,000
Test	1,320	660	660	2,640	900	450	450	1,800
Total	3,380	1,690	1,690	6,760	2,400	1200	1,200	4,800

Table 15: CLEF-PAN 2019: Number of authors per language and subset. .

Language	Training	Test	Total
English	300	200	500
Spanish	300	200	500

Table 16: CLEF-PAN 2020: Number of authors in the training and test set.

speech." Finally, two hundred tweets per Twitter user were gathered to create the final dataset. Table 35 contains the distribution within this corpus.

Language	Training	Test	Total
English	200	100	300
Spanish	200	100	300

Table 17: CLEF-PAN 2021: Number of authors in the training and test set.

4.4 Experiments and Results

4.4.1 Data Collection

We did not label our own data for this thesis because we found a substantial amount of labelled data with verified baselines through our involvement in CLEF-PAN evaluation campaigns. We utilize them as a benchmark for the proposed strategies. For author profiling tasks, we use CLEF-PAN 2013, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021. We have data on gender, age range group, language variation, hate speech spreaders, fake news spreaders, and bots from the list sets. Table 18 up to Table 23 below provide examples of text data for the classes in these categories.

	English tweets
MALE	Funny how the recent democrats primary elections here had an unexpected scientific side effect: u can't predict anything with social media. @username what if the kids were screaming at you barking dog ? perspectives ; I'm becoming super skilled at lighting the fire!. This is a little bit technical but if you work with online sns and sna you might find it interesting. I am watching Dexter, Sin of Omission (S06E08). y bitch stay my bitch so don't look she keep that pussy tucked. I ate shit tryna dunk playing ball earlier. It's cool though I'm still a mufucking balla fur real. First day of work today. Kinda nervous but mostly excited.
FEMALE	For fitness day, I made my tots run lines, do sit-ups, and race through an obstacle course in which the obstacles were babies. #daycarelife. @username bahaha! unfortunately i'm not a tshirt person. After 3 months of beating my head against a wall over this persuasion, I've finally had my "aha!" moment!! progress! improvement! The people you choose to work with are the most accurate predictor of job satisfaction I've ever found. I am definitely at the upper limit of normal. Every choice, no matter how small, begins a new story. We ain't picture perfect but we worth the picture still. @username: Don't be so quick to believe what you hear because lies spread quicker than the truth. I wish I could make it easy, easy to love me.

Table 18: Sample of three tweets in English for each class of gender identification

	English tweets
18 - 24	I wish we could talk forever. 100000 things I want to say... But it's better I keep my mouth shut. Those dreams where the feeling you have is the best thing. Nah Puyol would get a jab in the lip for that ???????? @username: So now tall ppl ain't approachable ? I joss don't kno man"don't worry bro it's the beard too???? all and beard gang = Unapproachable, Short and Baby face = Bae ????
25 - 34	How to Test Your Startup Idea for $50. Business models are a commodity Stop asking "But how will they make money? I don't want to archieve immortality through my work... I want to achieve it through not dying." Woody Allen. Made south western lasagna and a strawberry pie last night. texasliving Can't wait for my paycheck!!! I'm jonsing for some pizza hard core!! stonergoals My kingdom for a New York slice. munchies homesick pizza ???????? #eastcoast
35 - 49	I love how you write! I wish I could have half of your talent to express my feelings in writing. Remembre it only takes ONE negative comment to kill a dream". Think about that when someone share it with you.Kindness is the language which the deaf can hear and the blind can see" Mark Twain @username kisses....thanks for watching the show. And happy bday to you... @username @username doña Catherine @username sube al cielo junto a beatrice, julieta y venus...no williams. @username y @username en Agosto. Unforgettable acting. Actuaciones inolvidables. Thank you...
50+	@username please check your twitter DM box for your photoedit @username @username @username @username @username @username Nikki is a forever young baby! ???? username just wished me a Happy belated Birthday! Thank you very much, Nita! ???????????????? username: Putin, Assad, and Obama are playing a high-stakes poker game, and @username explains the rules @username: "religion wants us to believe without a deep understanding of things" @username #CDIdeas #ideaspeligrosas #fail @username Jack Ruby shot L. Harvey Oswald 63, a photographer almost took the shot of a lifetime http://t.co/TWPt06dgGY"// gran dcto.!

Table 19: Sample of three tweets in English for each class of age range

	Portuguese tweets
Brazil	laystofmykind começa mesmo e vai me dando os feedback, haha . @laystofmykind amo quando associam naruto a mim, haha! Pois assiste pode começar, queria ser vc só pra poder ver tudo de novo. Eu disse que ia ver BBB e perde nada .. hahaha desde que começou eu nao assistir 1 episodio se quer, sei nem quem é essas do paredão ai . Meu pai e suas invenções ,já tá inventando de ir pra águas lindas amanhã Bateu saudade de uns tempos atrás @lipemct sei muito bem essa amizade.
Portugal	PARABÉNS YOU BITCH, hope you have an amazin' day longe dazamiga. Mereces este mundo e os outros q ainda estão p des... Não vou ao Dragão ver o Sporting por causa dos anos da Rita.....onde é que isto já se viu Tenho nariz todo assado, pareço o Rodolfo Isto do PresidentBannon é giro! Só queria acrescentar mais uma contagem à hash tag. hugoloureiro7 Obrigado. não me apetece explicar, deixo o link: Alguém me consegue dizer se o Enzo Perez para assinar pelo Benfica é aquele que já passou por cá, ou é outro, o do Valência? CarregaBenfica @pipaa28 boa noite, até amanhã , @cervi_franco hoje vais marcar golo Chuky !!! Vamosss!!!! Juntos CarregaBenfica], HOJE É DIA DE BENFICA Rumo ao Tetra !!!

Table 20: Sample of three tweets in Portuguese for each variety

	bots vs human tweets
bot	Backend Developer: Responsibilities * Develop the core infrastructure, framework and services that incorporate our proprietary algorithms * Collaborate closely with PM, UX, and other stakeholders to clarify functional flows and provide technical insights... Senior Symfony (PHP) Developer: We are seeking a highly-talented Programmer / Developer who will be responsible for creating beautiful, engaging web experiences for high-end clients. We are looking for an innovative thinker who is fully committed to web... Director/Engineering Supervisor: Director, Global Certification (Bachelor + 5 years progressive experience), Engineering Supervisor Body Structures (Bachelor + 5 years progressive experience) sought by Karma... Administrative Assistant/Data Encoder: We are seeking a full time and part time Administrative Assistant/Data Entry. You will perform clerical,administrative clerical support and personal assistant functions....
human	Never ever ordering from @jeffreyboutique again still not sent my dress told me they'd give me a refund and no had that either @robbieurq1 @beckyfrancesxo @glasgow_sophie @karmen1998 Dinny speak to my pal like that @ionaagriffin Yessss but that's tomoz I want out now !! tryin to save myself If salespeople think of what they do as at odds with who they are or what they want to achieve in life, they will fail. If they are comfor... 3 Things Every College Grad Should Keep In Mind as They Look for a... @sambrabender @PatsPartyRental @LisaRots @garydmclean That's exciting!

Table 21: Sample of three tweets in English for each class of bots and human

	fake news spreader tweets
0	RT USER: People don't think La Liga players try to kick Messi? Did you miss the CL final when peak Vidic; Rio tried to assault him and... This dunk contest, Gordon had the crazier dunks but Zach Levine won. Both were super but I feel Gordon was on a dif... URL RT USER: 2016: Three straight 50s after the first round, lost on the second dunk-off. 2020: Five straight 50s, lost on the second du... This global framework recognises the skills and experience of HASHTAG professionals no matter their skill leve... URL Is there still a stigma when it comes to HASHTAG despite society's' awareness of their advantages? Kirstie... URL Our Functional Skills qualifications help individuals gain confidence through their learning and work journey. Fin... URL
1	Scientists just discovered that an asteroid may have ended 'Snowball Earth' 2.2 billion years ago... URL Yang Says He Would 'Pardon Trump'... There's Just One Problem With That URL URL 10 can't-miss sales to shop this weekend at The North Face, Wayfair, and more URL URL oe Biden Wants To Destroy Free Speech on Social Media URL Dog Holds His New Dad's Arm On His Way Home, Refuses To Let Go URL Others Said He Was Beyond Saving So Let Him Die Alone, They Didn't Listen URL

Table 22: Sample of three tweets in English for each class of fake news spreaders

	hate speech spreader tweets
0	RT USER: Funny how "15 days to slow the spread" turned into "maybe you can have a barbecue in July of 2021." RT USER: Why did Minneapolis just give George Floyd's family $27 million for overdosing on fentanyl while he resisted arrest? RT USER: To be fair, he has done a lot of undercover work on Chinese communists. URL If you havent tried Parler yet please give it a go. Its awesome!! URL RT USER: Attention Small Business Owners: Senate Democrats Blocked Your Relief Bill URL RT USER: Is now a good time for Nancy and the Dems to try and impeach Trump again?
1	USER Shittf needs to be on the chopping block right next to hrc! Off With Their Heads!!! Just bought an awesome PX-60 keyboard at Bananas At Large and am LOVIN' it! USER] USER USER According to who? That's hilarious! RED TSUNAMI COMING YOUR WAY! URL USER Take a bow Cynthia philips along with other teachers who take a interest in their pupils well being of all ages. Can we have a immigration system with equal religions and cultures not based on one religion please we are all at war with Covid USER USER Doh o dear a female dear a bit of jam and bread wil bring you back to doh a dear

Table 23: Sample of three tweets in English for each class of hate speech spreaders

4.4.2 Preprocessing

All the data sets received the same treatment during preprocessing. In the following order, conversion to lowercase, text normalization, text stemming, all url were changed to " urllink ". Contractions were changed to a one word i.e. don't (do + not) becomes donot, can't (can + not) cannot.

Before selecting the tokens, we also removed words that only appeared one time in a text collection i.e. words that appeared only once in a male text or female message were removed. They correspond often to spelling errors.

4.4.3 Feature Selection and Text Representation

A decrease in the number of features is required to be able to focus on the crucial stylistic markers that can discriminate between men and women, age range groups, language variety.

The different document text representation were computed as described in Section 3.2. For each feature selected, the text is represented as the term frequency in the document divided by the length of the document. This will ensure that the presence of the token is not determined by the size of the document.

The first set of features are extracted using the the two-stage feature extraction technique. From the extracted features, we apply further feature reduction by ranking these selected features using Chi-square (χ^2) Section 2.2.6 and Pointwise Mutual Information (PMI) scoring techniques (Section 2.2.4). Only the top 300 features generated by these methods are then used to create the other models. For each dataset, we will therefore have three models for each feature selection.

4.4.4 Classification

Following the scikit-learn library with default parameters for all the experiments, we used KNN, SVM, ExtraTrees, Decision Tree, Gaussian NB, Bernoulli NB, Multinomial NB, MLP, SGD, LDA, Random Forest, AdaBoost, Bagging, Gradient Boosting, XGB, and Logistic Regresion. The choice of applying the default parameter values is required e.g. the fact that faced with a new corpus the default values are the entry points. Moreover, tuning hyperparameters to increase the accuracy for a given collection has limited value. This optimisation is usually different for another corpus.

Utilizing the three different feature sets provided in Section 4.4.3, three models are produced using each of the algorithms listed. For instance, KNN with all of the features from the two-stage feature selection technique, KNN with the top 300 (χ^2) ranked features, and KNN with the top 300 (PMI) ranked features.

When counting the number of experiments, we obtain the following values, the identification of fake news spreaders (6), gender (63), age range groups (39), language varieties (12), hate speech spreaders (6), and bots (6) will subsequently be done using these sets of models. There are 132 author profiling classifications in total for each algorithm and data samples.

The size of the corpus has a major impact on how well each strategy works. Furthermore, the findings show that none of the suggested author profiling approaches consistently outperforms the others. The performances are computed using only the test set as specified within each evaluation campaign.

Deep learning approaches have received a great deal of attention recently as a result of their success, particularly in image identification. In a similar spirit, word embedding has become popular recently in a number of projects involving natural language processing. We assessed the performance of four models CNN, LSTM, GRU, and CNN+LSTM utilizing unigram, bigram, and trigram of word features. However, these evaluations are limited to PAN 2014 and 2015 datasets (see Table 36 and 37)

Model	Feature	EN_g	EN_a
KNN	All	0.504	0.555*
	Chi2	0.503	0.555*
	PMI	0.501	0.554*
SVM	All	0.499	**0.639**
	Chi2	0.501	**0.639**
	PMI	0.501	0.632
ExtraTrees	All	0.499	0.638
	Chi2	0.499	0.638
	PMI	0.502	0.629
Decision Tree	All	0.503	0.511*
	Chi2	0.496	0.511*
	PMI	0.499	0.513*
Gaussian NB	All	0.499	0.589
	Chi2	0.500	0.589
	PMI	0.499	0.157
Bernoulli NB	All	0.501	0.599
	Chi2	0.498	0.599
	PMI	0.498	0.593
Multi. NB	All	0.499	0.575
	Chi2	0.502	0.575
	PMI	0.502	0.573
MLP	All	0.497	0.637
	Chi2	0.496	0.637
	PMI	0.500	0.625
SGD	All	0.501	0.622
	Chi2	0.499	0.622
	PMI	0.501	0.602
LDA	All	0.500	0.628
	Chi2	0.499	0.628
	PMI	0.496	0.620
Random Forest	All	0.500	0.632
	Chi2	0.500	0.632
	PMI	**0.505**	0.628
AdaBoost	All	0.495	0.611
	Chi2	0.502	0.611
	PMI	0.496	0.612
Bagging	All	0.503	0.581
	Chi2	0.500	0.581
	PMI	0.502	0.577
Gradient Boosting	All	0.499	0.633
	Chi2	0.501	0.633
	PMI	0.500	0.625
XGB	All	0.501	0.633
	Chi2	0.499	0.633
	PMI	0.499	0.625
Logistic Reg	All	0.500	0.625
	Chi2	0.500	0.625
	PMI	0.501	0.621

Table 24: CLEF-PAN 2013 gender and age range identification results.

		EN_b	ES_b	EN_r	EN_t	ES_s	EN_s
KNN	All	0.615	0.571	0.555*	0604*	0.558*	0.258*
	Chi2	0.615	0.536	0.611*	0.571*	0.580*	0.764
	PMI	0.641	**0.589**	0.569*	0.682	0.602*	0.262*
SVM	All	0.590	0.411*	0.668	0.578*	0.650	0.210*
	Chi2	0.577	0.500*	0.657	0.617*	0.647	0.248*
	PMI	0.628	0.500*	0.660	0.727	0.625	0.677*
ExtraTrees	All	0.615	0.482*	0.699	0.630*	0.631	0.253*
	Chi2	0.615	0.536	0.710	0.695	0.661	0.257*
	PMI	0.577	0.500*	0.646	0.643*	0.638	0.251*
Decision Tree	All	0.551*	0.429*	0.636	0.558*	0.553*	0.765
	Chi2	0.449 *	0.411*	0.627*	0.584*	0.620	0.764
	PMI	0.590	0.554	0.597*	0.630*	0.581*	0.256*
Gaussian NB	All	**0.654**	0.518	0.534*	0.578*	0.574*	0.747
	Chi2	0.603	0.571	0.599*	0.617*	0.562*	0.574*
	PMI	**0.654**	0.429*	0.559*	0.584*	0.525*	0.602*
Bernoulli NB	All	0.513*	0.464*	0.667	0.526*	0.580*	0.249*
	Chi2	0.526*	0.464*	0.707	0.532*	0.624	0.256*
	PMI	0.513*	0.446*	0.695	0.545*	0.571*	0.279*
Multi. NB	All	0.474*	0.411*	0.665	0.578*	0.569*	0.640*
	Chi2	0.487 *	0.464*	0.655	0.565*	0.606	0.650*
	PMI	0.500	0.518	0.653	0.636	0.620	**0.841**
MLP	All	0.615	0.518	0.669	0.675*	0.670	0.805
	Chi2	0.551*	0.446*	0.710	0.623*	0.631	0.206*
	PMI	0.615	0.500	0.678	0.500*	0.500*	0.500*
SGD	All	0.500*	0.500*	0.606*	0.558*	0.530*	0.517*
	Chi2	0.526*	0.500*	0.522*	0.500*	0.595*	0.517*
	PMI	0.487*	0.500*	0.534*	0.500*	0.500*	0.500*
LDA	All	0.577	0.464*	0.487 *	0.649*	0.599*	0.765
	Chi2	0.590	0.482*	0.681	0.545*	0.613	0.228*
	PMI	0.615	0.446*	0.640	0.571*	0.602*	0.758
Random Forest	All	0.577	0.482*	0.705	0.630*	0.654	0.251*
	Chi2	0.577	0.482*	0.710	0.708	0.677	0.251*
	PMI	0.564*	0.464*	0.657	0.669*	0.643	0.247*
AdaBoost	All	0.603*	0.500*	0.704	0.630	0.661	0.277*
	Chi2	0.538	0.464*	0.700	0.656*	0.666	0.239*
	PMI	0.551*	0.571	0.674	0.695	0.643	0.275*
Bagging	All	0.577	0.429	0.664	0682	0.640	0.218 *
	Chi2	0.551*	0.482	0.672	0.675*	0.666	0.700*
	PMI	0.551*	0.500	0.644	0.636*	0.592*	0.212*
Gradient Boosting	All	0.641	0.446	**0.724**	0.688	0.689	0.765
	Chi2	0.603	0.446	0.373*	0.766	0.670	0.238*
	PMI	0.538*	0.536	0.680	0.682	0.640	0.252*
XGB	All	0.628	0.500*	0.719	0.695	**0.693**	0.773
	Chi2	0.615	0.426*	0.719	**0.779**	0.671	0.243*
	PMI	0.577	0.482*	0.677	0.662*	0.629	0.257*
Logistic Reg	All	0.526*	0.429*	0.638	0.565*	0.606	0.297*
	Chi2	0.487*	0.482*	0.628*	0.558*	0.604	0.321*
	PMI	0.500*	0.554	0.602*	0.636*	0.608	0.773

Table 25: CLEF-PAN 2014 gender identification results.

Model	Feature	EN_b	ES_b	EN_r	EN_t	ES_s	EN_s
KNN	All	0.397	0.250*	0.247*	0.318*	0.318	0.281*
	Chi2	**0.449**	0.339*	0.235*	0.370	0.325	0.275*
	PMI	0.321*	0.357	0.224*	0.383*	0.292	0.275*
SVM	All	0.308*	0.464	0.267*	0.377	0.316	0.325
	Chi2	0.244*	0.464	0.276	0.377	0.323	0.313
	PMI	0.359*	0.464	0.264	0.383	0.302*	0.305
ExtraTrees	All	0.308*	0.429	0.336	0.383	0.348	0.329
	Chi2	0.372*	0.429	0.314	0.357	0.336	0.318
	PMI	0.359*	0.429	0.283*	0.318*	0.300*	0.317
Decision Tree	All	0.256*	0.286*	0.242*	0.305*	0.274*	0.277*
	Chi2	0.282*	0.304*	0.244*	0.286*	0.292*	0.285*
	PMI	0.308*	0.375*	0.263*	0.247*	0.274*	0.270*
Gaussian NB	All	0.359*	0.446	0.236*	0.318 *	0.313	0.291*
	Chi2	0.256*	0.268*	0.184*	0.253*	0.290*	0.260*
	PMI	0.333*	0.32*1	0.147*	0.318*	0.111*	0.153*
Bernoulli NB	All	0.321*	0.464	0.283*	0.299*	0.345	0.284*
	Chi2	0.244*	0.286*	0.307	0.351	0.313	0.294*
	PMI	0.321*	0.464	0.297*	0.344*	0.314	0.279*
Multi. NB	All	0.308*	0.464	0.294*	0.377	0.318	0.290*
	Chi2	0.308*	0.464	0.284*	0.377	0.318	0.290*
	PMI	0.308*	0.464	0.255*	0.377	0.318	0.290*
MLP	All	0.333*	0.464	**0.353**	0.351	0.346	0.302
	Chi2	0.256 *	0.464	0.326	0.364	0.337	0.323
	PMI	0.308*	0.464	0.299*	0.377	0.318	0.313
SGD	All	0.436	0.196*	0.258 *	0.377	0.293*	0.275 *
	Chi2	0.423	**0.482**	0.244*	0.364	0.242*	0.305
	PMI	0.410	0.464	0.192*	0.377	0.244*	0.292*
LDA	All	0.308*	0.161*	0.201*	0.344*	0.201*	0.278*
	Chi2	0.321*	0.321*	0.305*	0.221*	0.318	0.316
	PMI	0.295*	0.286*	0.267*	0.143	0.288*	0.308
Random Forest	All	0.333*	0.286*	0.317	0.351	0.322	0.329
	Chi2	0.321*	0.393*	0.311	0.312*	0.323	0.324
	PMI	0.372*	0.446	0.302*	0.377	0.281*	0.315
AdaBoost	All	0.359*	0.429	0.308	0.383	0.341	0.300
	Chi2	0.359*	0.429	0.337	**0.396**	0.332	0.299
	PMI	0.359*	0.375*	0.309	0.331*	0.307*	0.303
Bagging	All	0.346*	0.304*	0.275*	0.312*	**0.359**	0.294*
	Chi2	0.295*	0.339*	0.255*	0.338	0.329	0.290*
	PMI	0.359*	0.375*	0.263*	0.370	0.292*	0.289*
Gradient Boosting	All	0.308*	0.357*	0.316	0.351	0.330	0.336
	Chi2	0.321*	0.411*	0.325	0.351	0.309*	0.315
	PMI	0.321*	0.411*	0.307	0.370	0.299*	0.315
XGB	All	0.321*	0.268*	0.340	0.370	0.334	**0.339**
	Chi2	0.295*	0.304*	0.339	0.344*	0.346	0.320
	PMI	0.333*	0.321*	0.305 *	0.377	0.300*	0.321
Logistic Reg	All	0.308*	0.464	0.270*	0.377	0.318	0.312
	Chi2	0.308*	0.464	0.262*	0.377	0.318	0.304
	PMI	0.308*	0.464	0.253*	0.377	0.318	0.294*

Table 26: CLEF-PAN 2014 age range identification results.

Model	Feature	EN	ES
KNN	All	0.486*	0.341
	Chi2	0.465*	0.409
	PMI	0.423*	0.307
SVM	All	0.570	0.500
	Chi2	0.577	0.500
	PMI	0.528*	0.500
ExtraTrees	All	0.620	0.477
	Chi2	0.592	0.466
	PMI	0.570	0.352*
Decision Tree	All	0.479*	0.330*
	Chi2	0.472*	0.420*
	PMI	0.451*	0.341*
Gaussian NB	All	0.444*	0.284*
	Chi2	0.458*	0.432*
	PMI	0.169*	0.205*
Bernoulli NB	All	**0.655**	**0.500**
	Chi2	**0.655**	**0.500**
	PMI	0.620	**0.500**
Multi. NB	All	0.444*	**0.500**
	Chi2	0.451*	**0.500**
	PMI	0.423*	**0.500**
MLP	All	0.599	0.466
	Chi2	0.613	0.500
	PMI	0.535*	0.500
SGD	All	0.373*	0.443
	Chi2	0.275*	0.500
	PMI	0.437*	0.205*
LDA	All	0.472*	0.386*
	Chi2	0.423*	0.375*
	PMI	0.458*	0.398*
Random Forest	All	0.606	0.489
	Chi2	0.613	0.443
	PMI	0.556*	0.330*
AdaBoost	All	0.507*	0.364*
	Chi2	0.408*	0.364*
	PMI	0.451*	0.409*
Bagging	All	0.613	0.386*
	Chi2	0.613	0.455
	PMI	0.648	0.330*
Gradient Boosting	All	0.514*	0.398*
	Chi2	0.521*	0.420*
	PMI	0.472*	0.364*
XGB	All	0.542*	0.466
	Chi2	0.599	0.420*
	PMI	0.486*	0.420*
Logistic Reg	All	0.556*	0.500
	Chi2	0.542*	0.500
	PMI	0.486*	0.500

Table 27: CLEF-PAN 2015 age range identification results.

Model	Feature	EN	ES	IT	NL
KNN	All	0.775	0.818	0.556*	0.656*
	Chi2	0.746	0.830	0.556*	0.625*
	PMI	0.711	0.761*	0.694	0.688
SVM	All	0.746	0.818	0.583*	0.719
	Chi2	0.676*	0.773*	0.556*	0.688
	PMI	0.746	0.818	0.667	0.594*
ExtraTrees	All	0.761	0.875	**0.694**	0.719
	Chi2	**0.796**	**0.909**	**0.694**	0.750
	PMI	0.746	0.886	0.667	0.688
Decision Tree	All	0.634*	0.682*	0.611	0.719
	Chi2	0.676*	0.784*	0.500*	0.656*
	PMI	0.570*	0.784*	0.639	0.719
Gaussian NB	All	0.732	0.852	0.611	**0.781**
	Chi2	0.782	0.886	0.611	**0.781**
	PMI	0.690*	0.682*	0.694	0.688
Bernoulli NB	All	0.676*	0.807	0.667	0.750
	Chi2	0.697	0.875	0.611	0.750
	PMI	0.711	0.773*	0.556*	0.594*
Multi. NB	All	0.620*	0.568*	0.611	0.500*
	Chi2	0.620*	0.591*	0.611	0.531*
	PMI	0.528*	0.580*	0.639	0.500*
MLP	All	0.754	0.875	0.639	0.750
	Chi2	0.683*	0.818	0.611	0.688
	PMI	0.500*	0.500*	0.639	0.625*
SGD	All	0.683*	0.500*	0.500*	0.500*
	Chi2	0.500*	0.500*	0.500*	0.500*
	PMI	0.500*	0.500*	0.500*	0.500*
LDA	All	0.627*	0.739*	0.472*	0.531*
	Chi2	0.754	0.727*	0.556*	0.406*
	PMI	0.634*	0.659*	0.667	0.406*
Random Forest	All	0.761	0.898	0.583*	0.781
	Chi2	0.789	0.864	0.639	0.750
	PMI	0.775	0.784*	0.528*	0.562*
AdaBoost	All	0.725	0.716*	0.611	0.656*
	Chi2	0.683*	0.693*	0.611	0.562*
	PMI	0.577*	0.727*	0.694	0.562*
Bagging	All	0.655*	0.750*	0.583*	0.688
	Chi2	0.704	0.784*	0.639	0.500*
	PMI	0.732	0.773*	0.583*	0.531*
Gradient Boosting	All	0.754	0.784*	0.583*	0.719
	Chi2	0.718	0.795	0.583*	0.719
	PMI	0.761	0.795	0.611	0.719
XGB	All	0.711	0.795	0.639	0.656*
	Chi2	0.704	0.784*	0.639	0.594*
	PMI	0.725	0.784*	0.639	0.625*
Logistic Reg	All	0.592*	0.705*	0.639	0.500*
	Chi2	0.585*	0.682*	0.639	0.500*
	PMI	0.676*	0.818	0.667	0.500*

Table 28: CLEF-PAN 2015 gender identification results.

Model	Feature	EN_g	EN_a
KNN	All	0.500	0.256*
	Chi2	0.500*	0.372 *
	PMI	0.513 *	0.333*
SVM	All	0.500*	0.397*
	Chi2	0.500*	0.423
	PMI	0.564*	0.436
ExtraTrees	All	0.628	0.397*
	Chi2	0.628	0.385*
	PMI	0.641	0.385*
Decision Tree	All	0.590*	0.359*
	Chi2	0.590*	0.359*
	PMI	0.590*	0.410*
Gaussian NB	All	0.538*	0.397*
	Chi2	0.538*	0.256*
	PMI	0.564*	0.167*
Bernoulli NB	All	0.526*	0.333*
	Chi2	0.526*	0.295*
	PMI	0.538*	0.321*
Multi. NB	All	0.551*	0.410*
	Chi2	0.551*	0.410*
	PMI	0.500 *	0.410*
MLP	All	0.474*	**0.474**
	Chi2	0.474*	0.423
	PMI	0.500*	0.410*
SGD	All	0.679	0.321*
	Chi2	0.679	0.333*
	PMI	0.500*	0.346*
LDA	All	0.551*	0.397*
	Chi2	0.551*	0.462
	PMI	0.564*	0.385*
Random Forest	All	0.641	0.410*
	Chi2	0.641	0.385*
	PMI	0.654	0.372*
AdaBoost	All	0.641	0.397*
	Chi2	0.641	0.346*
	PMI	0.603*	0.282
Bagging	All	0.679	0.397*
	Chi2	0.679	0.385*
	PMI	0.628	0.397*
Gradient Boosting	All	**0.705**	0.436
	Chi2	**0.705**	0.423
	PMI	0.641	0.385*
XGB	All	0.654	0.449
	Chi2	0.654	0.346*
	PMI	0.615	0.385*
Logistic Reg	All	0.500	0.410*
	Chi2	0.500	0.410*
	PMI	0.500	0.410*

Table 29: CLEF-PAN 2016 gender and age range identification results.

Model	Feature	EN	ES	PT	AR
KNN	All	0.670*	0.610*	0.671*	0.620*
	Chi2	0.656*	0.607*	0.666*	0.639*
	PMI	0.679*	0.613*	0.686*	0.624*
SVM	All	0.760	0.725	0.746	0.688
	Chi2	0.745*	0.705	0.736*	0.665*
	PMI	**0.873**	0.679*	0.755	0.651*
ExtraTrees	All	0.754*	0.735	0.775	0.753
	Chi2	0.775	0.749	0.823	0.772
	PMI	0.766	0.714	0.786	0.718
Decision Tree	All	0.650*	0.624*	0.745	0.672*
	Chi2	0.672*	0.610*	0.746	0.659 *
	PMI	0.670*	0.624*	0.744	0.649*
Gaussian NB	All	0.664*	0.667*	0.709*	0.683
	Chi2	0.727*	0.704	0.726*	0.644
	PMI	0.640*	0.647*	0.718*	0.616
Bernoulli NB	All	0.731*	0.700	0.718*	0.678*
	Chi2	0.757*	0.705	0.760	0.719
	PMI	0.749*	0.697	0.769	0.686
Multi. NB	All	0.697*	0.668 *	0.664 *	0.634*
	Chi2	0.698*	0.657*	0.676*	0.630*
	PMI	0.667*	0.655*	0.589*	0.629*
MLP	All	0.799	**0.789**	0.810	0.777
	Chi2	0.792	0.757	0.784	0.731
	PMI	0.754*	0.714	0.733*	0.674*
SGD	All	0.661*	0.640*	0.685*	0.501*
	Chi2	0.639*	0.607*	0.536*	0.552*
	PMI	0.655*	0.589*	0.551*	0.500*
LDA	All	0.705*	0.661*	0.685*	0.520*
	Chi2	0.790	0.750	0.806	0.731
	PMI	0.738*	0.694	0.786	0.680
Random Forest	All	0.770	0.721	0.780	0.750
	Chi2	0.775	0.738	0.826	**0.781**
	PMI	0.772	0.706	0.815	0.728
AdaBoost	All	0.760	0.744	0.811	0.726
	Chi2	0.768	0.743	0.826	0.739
	PMI	0.744*	0.706	0.780	0.695
Bagging	All	0.726*	0.685*	0.802	0.742
	Chi2	0.741*	0.693	0.838	0.738
	PMI	0.725*	0.680*	0.815	0.710
Gradient Boosting	All	0.784	0.764	**0.844**	0.779
	Chi2	0.790	0.759	0.848	0.756
	PMI	0.770	0.712	0.818	0.727
XGB	All	0.792	0.774	0.840	0.771
	Chi2	0.783	0.755	0.848	0.762
	PMI	0.768	0.716	0.833	0.722
Logistic Reg	All	0.666*	0.642*	0.666*	0.611*
	Chi2	0.664*	0.642*	0.664*	0.608*
	PMI	0.655*	0.630*	0.691*	0.576*

Table 30: CLEF-PAN 2017 gender identification results.

Model	Feature	PT	ES	EN	AR
KNN	All	0.915	0.310*	0.280*	0.526*
	Chi2	0.941	0.318*	0.272*	0.537*
	PMI	0.951	0.444*	0.354*	0.617*
SVM	All	0.965	0.701*	0.504 *	0.714
	Chi2	0.964	0.634*	0.480*	0.689*
	PMI	0.970	0.770*	0.637*	0.722
ExtraTrees	All	0.978	0.912	0.777	0.794
	Chi2	0.979	**0.932**	**0.855**	0.798
	PMI	0.980	0.922	0.837	0.788
Decision Tree	All	0.927	0.782*	0.700 *	0.691*
	Chi2	0.938	0.812	0.715*	0.704
	PMI	0.931	0.810*	0.715*	0.689*
Gaussian NB	All	0.946	0.210*	0.262*	0.302*
	Chi2	0.975	0.834	0.689*	0.733
	PMI	0.976	0.808*	0.630*	0.713
Bernoulli NB	All	0.980	0.837	0.676*	0.724
	Chi2	0.980	0.895	0.776	0.761
	PMI	0.984	0.889	0.797	0.761
Multi. NB	All	0.961	0.396 *	0.408*	0.569*
	Chi2	0.958	0.371*	0.364*	0.531*
	PMI	0.926	0.432*	0.375*	0.622*
MLP	All	0.974	0.924	0.787	0.804
	Chi2	0.970	0.892	0.757	0.766
	PMI	0.968	0.895	0.734*	0.744
SGD	All	0.804*	0.254*	0.236*	0.369*
	Chi2	0.921	0.267*	0.255 *	0.422*
	PMI	0.756*	0.159*	0.206*	0.251*
LDA	All	0.971	0.754*	0.527*	0.249*
	Chi2	0.975	0.894	0.810	0.774
	PMI	0.973	0.885	0.765	0.757
Random Forest	All	0.980	0.915	0.812	0.808
	Chi2	0.978	0.910	0.841	0.801
	PMI	0.979	0.907	0.829	0.791
AdaBoost	All	0.978	0.833	0.786	0.717
	Chi2	**0.984**	0.841	0.802	0.747
	PMI	0.980	0.830	0.799	0.746
Bagging	All	0.953	0.847	0.773	0.744
	Chi2	0.949	0.860	0.785	0.757
	PMI	0.958	0.850	0.793	0.755
Gradient Boosting	All	0.981	0.906	0.845	0.804
	Chi2	0.980	0.915	0.848	0.804
	PMI	0.976	0.910	0.839	0.799
XGB	All	0.981	0.919	0.848	**0.809**
	Chi2	0.983	0.921	0.848	0.800
	PMI	0.979	912	0.846	0.797
Logistic Reg	All	0.940	0.285’	0.318*	0.454*
	Chi2	0.938	0.269*	0.295*	0.448*
	PMI	0.911	0.229*	0.263*	0.521*

Table 31: CLEF-PAN 2017 language variety identification results.

Model	Feature	EN	ES	AR
KNN	All	0.668*	0.597*	0.581*
	Chi2	0.657*	0.604*	0.584*
	PMI	0.658*	0.600*	0.599*
SVM	All	0.747	0.722	0.691
	Chi2	0.737	0.695	0.684
	PMI	0.724	0.680	0.661*
ExtraTrees	All	0.756	0.736	0.759
	Chi2	0.779	0.742	**0.780**
	PMI	0.752	0.695	0.690
Decision Tree	All	0.654*	0.594*	0.678
	Chi2	0.639*	0.610*	0.681
	PMI	0.656*	0.695	0.627*
Gaussian NB	All	0.661*	0.653*	0.682
	Chi2	0.713	0.701	0.669*
	PMI	0.641*	0.633*	0.634*
Bernoulli NB	All	0.723	0.702	0.708
	Chi2	0.757	0.708	0.737
	PMI	0.732	0.681	0.696
Multi. NB	All	0.687*	0.668*	0.632*
	Chi2	0.694	0.652*	0.639*
	PMI	0.591*	0.627*	0.601*
MLP	All	**0.794**	**0.782**	0.769
	Chi2	0.784	0.748	0.731
	PMI	0.742	0.685	0.671*
SGD	All	0.667*	0.588*	0.634*
	Chi2	0.606*	0.608*	0.588*
	PMI	0.650*	0.527*	0.584*
LDA	All	0.729	0.692	0.683
	Chi2	0.773	0.738	0.734
	PMI	0.733	0.685	0.658*
Random Forest	All	0.755	0.720	0.750
	Chi2	0.767	0.729	0.776
	PMI	0.761	0.696	0.699
AdaBoost	All	0.759	0.716	0.726
	Chi2	0.744	0.711	0.724
	PMI	0.737	0.687	0.666*
Bagging	All	0.715	0.680	0.728
	Chi2	0.725	0.694	0.736
	PMI	0.718	0.670*	0.689
Gradient Boosting	All	0.772	0.765	0.769
	Chi2	0.784	0.753	0.760
	PMI	0.756	0.709	0.682
XGB	All	0.774	0.773	0.760
	Chi2	0.782	0.755	0.770
	PMI	0.754	0.708	0.676*
Logistic Reg	All	0.666*	0.648*	0.604*
	Chi2	0.668*	0.644*	0.589*
	PMI	0.657*	0.615*	0.564*

Table 32: CLEF-PAN 2018 gender identification results.

Model	Feature	EN_g	ES_g	EN_b	ES_b
KNN	All	0.539*	0.574*	0.896	0.832
	Chi2	0.580*	0.549*	0.902	0.851
	PMI	0.598*	0.606*	0.905	0.853
SVM	All	0.688*	0.668	0.931	**0.925**
	Chi2	0.689*	0.637	0.924	**0.925**
	PMI	0.681*	0.624	0.925	0.914
ExtraTrees	All	0.748	0.673	0.905	0.849
	Chi2	0.751	0.680	0.928	0.885
	PMI	0 .759	0.687	0.919	0.886
Decision Tree	All	0.656*	0.583*	0.916	0.837
	Chi2	0.673*	0.567*	0.898	0.804*
	PMI	0.627*	0.599*	0.904	0.797*
Gaussian NB	All	0.643*	0.570*	0.811*	0.731*
	Chi2	0.656*	0.543*	0.787*	0.736*
	PMI	0.660*	0.554*	0.823	0.798*
Bernoulli NB	All	**0.805**	0.668	0.870	0.843
	Chi2	0.775	0.664	0.892	0.848
	PMI	0.742	0.654	0.876	0.844
Multi. NB	All	0.655*	0.612*	0.829	0.823
	Chi2	0.652*	0.614*	0.838	0.825
	PMI	0.624*	0.624	0.852	0.824
MLP	All	0.774	**0.716**	0.864	0.867
	Chi2	0.739	0.664	0.901	0.879
	PMI	0.718	0.646	0.882	0.828
SGD	All	0.632*	0.564*	0.891	0.842
	Chi2	0.621*	0.524*	0.889	0.856
	PMI	0.573*	0.548*	0.883	0.846
LDA	All	0.730	0.633	0.579*	0.647*
	Chi2	0.748	0.632	0.866	0.836
	PMI	0.711	0.612*	0.838	0.846
Random Forest	All	0.734	0.697	0.926	0.876
	Chi2	0.741	0.696	0.939	0.896
	PMI	0.755	0.692	0.938	0.888
AdaBoost	All	0.712	0.660	0.924	0.872
	Chi2	0.730	0.651	0.930	0.871
	PMI	0.717	0.656	0.920	0.872
Bagging	All	0.692*	0.619*	0.927	0.882
	Chi2	0.691*	0.620*	0.924	0.862
	PMI	0.702	0.633	0.915	0.866
Gradient Boosting	All	0.746	0.697	0.938	0.887
	Chi2	0.758	0.678	0.942	0.892
	PMI	0.725	0.660	0.939	0.883
XGB	All	0.752	0.711	**0.944**	0.894
	Chi2	0.742	0.658	0.937	0.893
	PMI	0.728	0.660	0.942	0.891
Logistic Reg	All	0.620*	0.570*	0.891	0.843
	Chi2	0.617*	0.572*	0.890	0.843
	PMI	0.591*	0.583*	0.890	0.844

Table 33: CLEF-PAN 2019 bot and gender identification results.

Model	Feature	EN	ES
KNN	All	0.555*	0.710
	Chi2	0.555*	0.730
	PMI	0.580 *	0.750
SVM	All	0.620*	0.700
	Chi2	0.610*	0.705
	PMI	0.550*	0.720
ExtraTrees	All	0.695	0.740
	Chi2	0.700	**0.760**
	PMI	0.700	0.755
Decision Tree	All	0.590*	0.660*
	Chi2	0.610*	0.715
	PMI	0.595*	0.715
Gaussian NB	All	0.630*	0.705
	Chi2	0.645*	0.735
	PMI	0.665*	0.670
Bernoulli NB	All	0.715	0.705
	Chi2	0.705	0.740
	PMI	0.690	0.700
Multi. NB	All	0.555*	0.685
	Chi2	0.565*	0.685
	PMI	0.590*	0.685
MLP	All	0.725	0.725
	Chi2	0.645*	0.700
	PMI	0.505*	0.695
SGD	All	0.610*	0.560*
	Chi2	0.500*	0.500*
	PMI	0.500*	0.525*
LDA	All	0.710	0.720
	Chi2	0.560 *	0.515*
	PMI	**0.785**	0.580*
Random Forest	All	0.730	0.735
	Chi2	0.710	0.755
	PMI	0.670*	0.715
AdaBoost	All	0.645*	0.715
	Chi2	0.635*	0.745
	PMI	0.640*	0.675
Bagging	All	0.675 *	0.720
	Chi2	0.650*	0.705
	PMI	0.660 *	0.740
Gradient Boosting	All	0.660*	0.740
	Chi2	0.655*	0.745
	PMI	0.620*	0.740
XGB	All	0.680*	0.730
	Chi2	0.695	0.730
	PMI	0.650*	0.725
Logistic Reg	All	0.620*	0.660*
	Chi2	0.620*	0.655*
	PMI	0.520*	0.685*

Table 34: CLEF-PAN 2020 fake news spreaders results.

Model	Feature	EN	ES
KNN	All	0.480*	0.740
	Chi2	0.480*	0.730
	PMI	0.610	0.730
SVM	All	0.500*	0.690*
	Chi2	0.470*	0.690*
	PMI	0.540*	0.69*
ExtraTrees	All	0.620	0.790
	Chi2	0.600	0.790
	PMI	0.580*	0.800
Decision Tree	All	0.530*	0.580*
	Chi2	0.460*	0.650*
	PMI	0.540*	0.680*
Gaussian NB	All	0.560*	0.800
	Chi2	0.620	0.790
	PMI	0.550	0.770
Bernoulli NB	All	**0.680**	0.780
	Chi2	**0.680**	0.800
	PMI	0.620	0.780
Multi. NB	All	0.510 *	0.700*
	Chi2	0.510*	0.690*
	PMI	0.510*	0.770
MLP	All	0.530*	0.800
	Chi2	0.540*	0.790
	PMI	0.550*	0.750
SGD	All	0.520*	0.610*
	Chi2	0.490*	0.520*
	PMI	0.490*	0.510*
LDA	All	0.660	0.790
	Chi2	0.540*	0.710*
	PMI	0.570*	0.740
Random Forest	All	0.560*	**0.83**0
	Chi2	0.560*	0.780
	PMI	0.610	0.810
AdaBoost	All	0.520*	0.780
	Chi2	0.580*	0.740
	PMI	0.560*	0.800
Bagging	All	0.560*	0.750
	Chi2	0.500*	0.770
	PMI	0.520*	0.790
Gradient Boosting	All	0.560*	**0.830**
	Chi2	0.530*	**0.830**
	PMI	0.560*	0.780
XGB	All	0.530*	0.810
	Chi2	0.520*	0.790
	PMI	0.600	0.800
Logistic Reg	All	0.530*	0.660*
	Chi2	0.530*	0.670*
	PMI	0.510*	0.660*

Table 35: CLEF-PAN 2021 hate speech spreaders results.

Model	Feature	EN_b	ES_b	EN_r	EN_s	ES_s
Conv1D	unigram	0.6026	0.5179	**0.7180**	**0.6282**	**0.6113**
	bigram	0.5641	**0.6071**	0.6876	0.5305	0.6025
	trigram	0.5385	0.5536	0.6108	0.5347	0.5813
LSTM	unigram	0.6026	0.5536	0.6236	0.5641	0.5265
	bigram	0.5513	0.5536	0.5914	0.5212	0.5548
	trigram	0.5513	0.5893	0.6121	0.5198	0.5265
Conv1D + LSTM	unigram	0.5256	0.5536	0.6224	0.5385	0.5265
	bigram	**0.6154**	0.5357	0.6169	0.5151	0.5477
	trigram	0.5128	0.5357	0.6139	0.5178	0.5265
GRU	unigram	0.5128	0.5179	0.5743	0.5190	0.5212
	bigram	0.5256	0.5714	0.5579	0.5113	0.5424
	trigram	0.5385	0.5179	0.6090	0.5133	0.5389

Table 36: CLEF-PAN 2014 deep learning gender identification results.

Model	Feature	EN	ES	NL	IT
Conv1D	unigram	0.7394	0.7727	**0.8125**	0.6944
	bigram	0.7394	0. 7955	**0.8125**	0.6111
	trigram	**0.7746**	**0.8068**	0.6562	0.6111
LSTM	unigram	0.6479	0.6705	0.5938	0.6389
	bigram	0.6901	0.6591	0.6875	0.5833
	trigram	0.6127	0.5341	0.6562	0.6667
Conv1D + LSTM	unigram	0.6549	0.7273	0.5938	**0.7222**
	bigram	0.6479	0.7386	0.6875	0.6111
	trigram	0.6620	0.6250	0.5938	0.6389
GRU	unigram	0.6408	0.5568	0.5938	0.6389
	bigram	0.6761	0.5909	0.6562	0.5833
	trigram	0.7183	0.6364	0.5938	0.5833

Table 37: CLEF-PAN 2015 deep learning gender identification results.

Model	Gender	Age range	Language	Fake News	Hate Speech	bot	Total
Number of Experiments	63	39	12	6	6	6	132
KNN	18	9	4	3	4	6	44
SVM	38	30	5	3	0	6	82
ExtraTrees	52	29	**12**	**6**	5	6	**110**
Decision Tree	19	2	5	2	0	4	32
Gaussian NB	24	8	6	3	4	1	46
Bernoulli NB	37	25	11	**6**	**6**	6	91
Multinomial NB	15	16	3	3	1	6	44
MLP	43	**32**	11	4	3	6	99
SGD	5	14	1	0	0	6	26
LDA	35	16	9	3	3	4	70
Random Forest	50	26	**12**	5	4	6	103
AdaBoost	42	24	**12**	3	3	6	90
Bagging	37	19	**12**	3	3	6	80
Gradient Boosting	**56**	24	**12**	3	3	6	104
XGB	51	23	**12**	4	4	6	100
Logistic Reg	14	17	3	1	0	6	41

Table 38: Analysis of performance of the algorithms

4.4.5 Overall Performance of the Models

Tables 24 through 35 presents the comparison of the proposed approach (two-step feature selection) against further feature reduction (PMI and χ^2) methods on the 44 authorship profiling datasets considered.

In Table 24 one can find the evaluation performed on year 2013 with both gender (EN_g) and age range (EN_a) for the English dataset. To simplify this evaluation, only one language was used (EN). Table 25 depicts the results datasets used in 2014 and are written in English (EN) and Spanish (ES). The text are coming from blogs (EN_b/ES_b), tweets (EN_t), reviews (EN_r) and social media (EN_s/ES_s). Similarly to CLEF-PAN 2013, in 2014 we have another author profiling task based on age range (Table 26).

In 2015, the gender identification involved four languages namely English (EN), Spanish (ES), Italian (IT) and Dutch (NL). Table 28 shows evaluation for gender identification while the age range identification results is in Table 27 two languages English (EN) and Spanish (ES) were considered.

For 2016 Table 29, shows gender (EN_g) and age range (EN_a) identification results in English (EN). In 2017 Table 30, depicts gender and Table 31, language variety identification results for four languages English (EN), Spanish (ES), Portuguese (IT) and Arabic (AR). In 2018 Table 32, reports results for gender identification in three languages English (EN), Spanish (ES) and Arabic (AR). In 2019 Table 33, presents results for a dataset in two languages English (EN), Spanish (ES) for bot (EN_b/ES_b) and gender (EN_g/ES_g) identification. Table 34, shows results for CLEF-PAN 2020 identification of fake news spreaders in two languages English (EN), Spanish (ES). For 2021 Table 35 reports results for identification of hate speech spreaders.

The performance achieved by different test collections cannot be directly compared. Additionally, it is impossible to specify an absolute average performance figure for a particular task. The accuracy distribution for the gender identification task illustrates performance variations in two gender identification datasets across the various languages as illustrated in Figure 11. One can see that for all experiments, the achieved performances are clearly above 0.5, a random baseline with a balanced dataset.

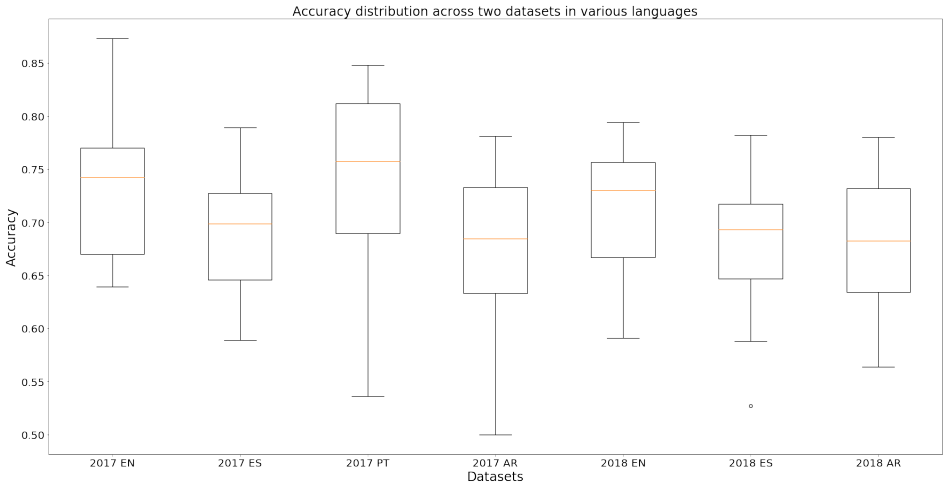


Figure 11: Accuracy distribution across datasets of two years in various languages

The sign test has been used to statistically determine whether one classification system should be seen as better than another scheme [50]. In all our tables, the best outcome depicted in bold has been compared to the accuracy rates. The null hypothesis H_0 asserts that both attribution models provide performance levels that are similar (significance threshold = 15%). In Tables 24 through 35, an asterisk (*) denotes differences that are statistically significant compared to the best performance.

The performance differences are not always statistically significant between the best system and the following ones. For example, as shown in Table 30, in 2017 EN gender prediction task the best performing system was a SVM using PMI features (0.873) and the ExtraTrees using PMI achieved an accuracy rate of 0.766, an absolute difference of 0.107 (or 13.97%). According to our statistical test, this variation was not significant. Similar to 2018 EN gender prediction with all features, Table 32, MLP had the highest accuracy rate of 0.794 while SVM produced a performance of 0.747 a performance difference of 0.047 (6.29%) was not significant.

Using results from Table 33 PAN 2019 EN gender prediction, we show cases with and without significant statistical difference in a graph as seen in Figure 12. The models with differences that are not statistically significant are those falling above the horizontal blue line (the threshold), while those with differences that are statistically significant fall below the horizontal line. For example ExtraTrees models with all three feature sets have differences that are not statistically significant from the best model (Bernoulli NB with all features at 0.805 accuracy).

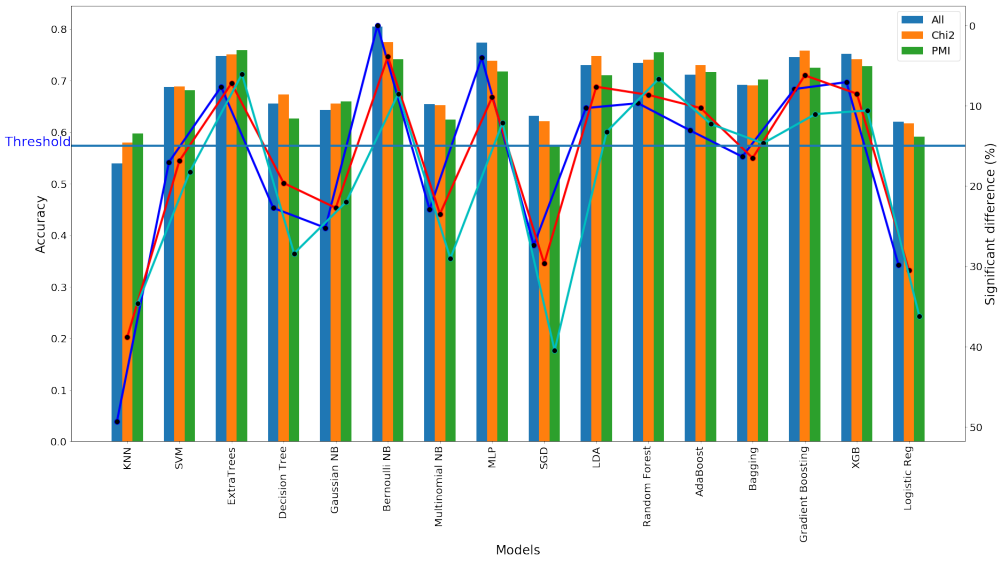


Figure 12: Statistically significant differences among the models

We may examine the 63 experiments in light of the underlying tasks to get a clearer perspective.

Gradient Boosting outperforms many times the other machine learning algorithms in scenarios involving gender identification. To be precise Gradient Boosting performs better 56 times over 63 models. To obtain an overview of all experiments, Table 38 indicates how many times a method offers better performance than another. Instead of considering the exact performance values, for each experiment we treat the subsets of models with performance differences that are not statistically significant as having the same performance. This subset contains all learning models presenting a performance statistically similar to the best one.

In a second place, one can find ExtraTrees (ocurring 52 times in the subset of the best performers). In the third position, we can see XGB (51 times), and then Random forest (50 times). In the lowest rank, Table 38 indicates SGD (5 times), Logistic regression (14) and Multinomial NB (15). Based on these results, we can form a set of best machine learning models with gradient boosting, ExtraTrees and XGB.

For age range detection, the best is MLP (32 times out of 39) followed by SVM (30 times) and in the third place is the ExtraTrees (29 times). Those with lowers ranks include KNN (9), Gaussian NB (8) and Decision Tree (2). This result confirms the best method indicated previously.

For language variety detection, however, ExtraTrees, Random Forest (RF), Adaboost, Bagging, and Gradient Boosting scored the highest effectiveness (12 times out of 12). While SGD (1), Multinomial NB (3) and Logistic regression (3) came in the lower ranking.

ExtraTrees and Bernouli NB had the maximum efficacy 6 times out of 6 in the subset of the best performing models for predicting those disseminating false information. MLP (0), Logistic regression (1) and Decision tree (2) made it to the lower ranks.

The best classifier for identification of those who promote hate speech was Bernouli NB, making an appearance 6 times out of 6 in the subset of best performing models 38. This is followed by ExtraTrees (5), Gausian NB (4) and XGB (4) Least appearance was seen in Logistic regression (0), SGD (0), Decision Tree (0) and SVM (0).

For bot detection, most of the models appeared 6 times out of 6 in the subset of best performing models except Decision Trees (4), SGD (4) and Gaussian NB (1).

ExtraTrees has the best overall performance of 110 highest efficacy over 132 examples among all the author profiling tasks mentioned, followed by Gradient boosting with 104 and Random forest at 103 Table 38.

Model	2014 EN_b	2014 ES_b	2014 EN_r	2014 EN_s	2014 ES_s
Conv1D + LSTM bigram Gaussian NB PMI	0.6154 0.654				
Percentage difference	6.3				
Conv1D bigram KNN PMI		0.6071 0.589			
Percentage difference		-3.0			
Conv1D unigram Gradient boosting ALL			0.718 0.724		
Percentage difference			0.8		
Conv1D unigram XGB ALL				0.6282 0.693	
Percentage difference				10.3	
Conv1D unigram Multinomial NB PMI					0.6113 0.841
Percentage difference					37.6

Table 39: Percentage differences between machine learning and deep learning for gender identification CLEF-PAN 2014

Model	2015 EN	2015 ES	2015 NL	2015 IT
Conv1D trigram ExtraTrees Chi2	0.7746 0.796	0.8068 0.909		
Percentage difference	2.8	12.7		
Conv1D unigram Gaussian NB ALL			0.8125 0.781	
Percentage difference			-3.9	
Conv1D + LSTM unigram ExtraTrees ALL				0.7222 0.694
Percentage difference				-3.9

Table 40: Percentage differences between machine learning and deep learning for gender identification CLEF-PAN 2015

To have an idea about the performance of deep learning models we have used the gender identification task from CLEF-PAN 2014 (Table 36) and CLEF-PAN 2015 (Table 37). Tables 36 and 37 display a summary of the accuracy results for deep learning models. From these tables Covn1D was able to produce greatest accuracies for the distinct data sets followed by Covn1D+LSTM combination as highlighted in bold. The highest accuracy for the features utilized comes from the unigram feature in 5 out of the 9 experiments, followed by the bigram feature with 3 best accuracies and the trigram feature with 2 best accuracies. The large numbers of word n-grams make it more difficult for the model to learn good parameters for them. For the same reason, the trigram was not able to produce more top scores. In every situation, classical machine learning models perform better than deep learning models. As features we have considered the unigram representation which is based on individual words. In addition we have generated bigram of words and trigram of words as possible representation.

The percentage differences between machine learning and deep learning are displayed on Table 39 and Table 40. With 9 experiments, machine learning obtained 6 positive differences and only 3 cases where deeplearning was able to out perform machine learning. The highest percentage difference of 37.6 is seen in CLEF-PAN 2014 ES_s between Conv1D unigram and Multinomial NB PMI, followed by 12.7 in CLEF-PAN 2015 ES between Conv1D trigram and ExtraTrees Chi2 and 10.3 CLEF-PAN 2014 EN_s between Conv1D unigram and XGB ALL.

5 Author Verification

Authorship verification is the task of analysing the linguistic patterns of two or more texts to determine whether they were written by the same author or not. In practice, we have a set of texts written by a certain author and we should decide whether another text is also by that author by learning the difference or similarity of the writing styles of the two types of document pairs.

The set of all authors with their corresponding text samples is usually referred to as reference set, which is analysed regarding the writing style of the candidates in order to compare it to the writing style of the anonymous author. From this, a decision about the true author is made.

Authorship verification techniques has reached a high level of accuracy enabling applications in digital humanities, text forensics, cyber-security,

5.1 Introduction

The words people use and the way they structure their sentences is distinctive, and can often be used to identify the author of a particular work.

Authorship verification (AV) is a digital text forensics research that concerns itself with the question, whether two documents have been written by the same person. To solve this task, the system analyses the linguistic patterns of two or more texts to determine whether they were written by the same author or not. The main challenge is to extract writing style features from texts and use them to determine how close in style different documents are.

The author of a given text had to be revealed by identifying some of his or her stylistic individualism and comparing the profiles of two authors, just like in authorship attribution. [147] propose quantifying writing style in texts as a way to express the author's individuality. Emojis are used in the feature option for Twitter user verification [148]. [52] uses a huge amount of linguistic data, such as vocabulary, lexical patterns, syntax, semantics, information content, and item distribution, to recognize and verify the author of a text.

Analysis of anonymous emails for forensic investigations [63], verification of historical literature [75], ongoing authentication employed in cybersecurity [95], and identification of changes in writing styles with Alzheimer patients [55] are all viable applications of author verification.

5.2 State of the art

In the domain of computational stylometry, authorship verification is the use of linguistic style learning to determine whether or not a document was produced by a specific author.

To resolve this task, various general approaches have been proposed.

The impostors' technique [81] is a strategy that has been used for tackling the authorship verification problem. This method proposes employing a set of imposter documents gathered from an external source (e.g. the web) in such a way that the impostor documents have a topic in common with the known and unknown document.

In *the Profile-Based Method* [109], all of the known documents are combined into a single large document, which represents an author profile. The most common character n-grams are then generated from the combined documents. Simultaneously, the most common character n-grams from the questioned document are retrieved. After that, the created feature vectors are compared to one another

using a dissimilarity function. If the resulting dissimilarity score exceeds a predetermined level, then the known documents and the questioned document have the same author.

To differentiate an unfamiliar text from a set of known documents, *the unmasking method* [80] creates an SVM classifier. It then removes a certain number of the most significant attributes and repeats the process for all documents. If the decline in classification accuracy isn't significant, the unknown document was created by the author under investigation. The logic behind this strategy is that the classifier will always be able to discriminate between the texts at first. When the texts are by the same author, the differences will be concentrated on a few key elements, and when they are removed, the discrepancies will be larger. Texts by the same author will be difficult to discern after the elimination of some key elements, although it will remain quite straightforward to find other variations between them.

The compression-based approach [53] consists of compression algorithm and a similarity measure as well as a strategy to determine a threshold that serves as the authorship acceptance criterion. It does not involve explicitly defined features but rather delegates the feature engineering procedure to a compression algorithm (e.g. PPMd from the library SharpCompress). The similarity between the resulting compressed documents are then determined using the Compression-based Dissimilarity measure that measure dissimilarity in terms of the length of compressed documents.

In *the first-order verifier model* [75], the cosine similarity between TFIDF-normalized, character tetragram representations of two texts are calculated and a grid search on the calibration data is used to shift the obtained scores.

The semantic representations for two tweets were compared using a *Siamese LSTM neural network technique* [148], with the network input being the text and emojis collected from tweets and the output being the similarity result. The authors gathered and manually annotated a particular tweet dataset for 108 people with various interests, with a 61.56% accuracy rate.

5.3 Corpora

5.3.1 CLEF-PAN 2021: Cross-Domain Authorship Verification

The corpus consists of data obtained from fanfiction.net, a sharing platform for fanfiction that comes from various topical domains (or 'fandoms') [15] [73]. The contents are mainly fictional texts produced by non-professional authors in the tradition of a specific cultural domain (or 'fandom'), such as a famous author or a specific influential work. Fanfiction is now abundantly available on the internet, as the fastest growing form of online writing providing a platform for data collection. This corpus contains 52,590 text pairs (denoted problems) from which 27,823 pairs correspond to the same author and 24,767 are pairs written by two distinct persons. Each text excerpt contains, in mean, 2,200 word-tokens.

5.3.2 CLEF-PAN 2015: Cross-genre and Cross-topic Authorship Verification

The PAN CLEF 2015 had four test collections were built, each containing at least 200 problems (training + testing). In each collection, all the texts matched the same language but can be cross-topic or cross-genre and may differ significantly. This training corpus is divided into four sub corpora Dutch, English, Greek and Spanish, where each sub corpus consists of 100 problems.

There are differences between the sub-corpora for each language. In the English part, only one known document per problem is provided. In Dutch and Greek parts, the number of known documents per

problem vary, whereas, in the Spanish part, there are always four known documents per problem. The documents of Greek and Spanish parts are, on average, longer than those of the Dutch and English parts. For all languages, positive and negative instances are equally distributed as shown in Table 41

	Genre	Type	Problems	Documents	Avg. known documents	Avg.words document
Traning	**Dutch**	cross-genre	100	276	1.76	354
	English	cross-topic	100	200	1.00	366
	Greek	cross-topic	100	393	2.93	678
	Spanish	cross-genre	100	500	4.00	954
Testing	**Dutch**	cross-genre	165	452	1.74	360
	English	cross-topic	500	1000	1.00	536
	Greek	cross-topic	100	380	2.80	756
	Spanish	cross-genre	100	500	4.00	946
	$\sum$		1265	3701	1.93	641

Table 41: CLEF-PAN 2015 Corpus Details

5.3.3 CLEF-PAN 2014: Several languages/genres

The training corpus was comprised a set of author verification problems in several languages/genres. Each problem consists of some (up to five) known documents by a single person and exactly one questioned document. All documents within a single problem instance will be in the same language and within-problem documents are matched for genre, register, theme, and date of writing. The document lengths vary from a few hundred to a few thousand words. As depicted in Table 42 corpora in both training and evaluation sets are balanced with respect to the number of positive and negative examples.

	Language	Genre	Problems	Documents	Avg. known documents	Avg.words document
Traning	**Dutch**	Essay	96	268	1.8	412.4
	Dutch	Review	100	202	1.0	112.3
	English	Essay	200	729	2.6	848.0
	English	Novel	100	200	1.0	3,137.8
	Greek	Article	100	385	2.9	1,404.0
	Spanish	Article	100	600	5.0	1,135.6
Testing	**Dutch**	Essay	96	268	1.8	412.4
	Dutch	Review	100	202	1.0	112.3
	English	Essay	200	729	2.6	848.0
	English	Novel	100	200	1.0	3,137.8
	Greek	Article	100	385	2.9	1,404.0
	Spanish	Article	100	600	5.0	1,135.6
	$\sum$		796	2,575	2.2	1,714.9
	$\sum$		1,492	4,959	2.3	1,7415.0

Table 42: CLEF-PAN 2014 Corpus Details

5.3.4 CLEF-PAN 2013: Several languages/genres

The corpus for the author identification task of CLEF-PAN-2013 covers three languages: English, Greek, and Spanish. For each language there is a set of problems, where one problem comprises a set of documents of known authorship by the same author and exactly one document of questioned authorship.

The training corpus comprised 10 problems in English, 20 problems in Greek and 5 problems in Spanish as indicated in Table 43. The evaluation corpus is balanced over the three languages comprising 30 problems in English, 30 problems in Greek and 25 problems in Spanish.

In all cases, the distribution of positive and negative problems in each corpus (and every language-specific sub-corpus) was balanced.

The English part of the corpus (collected by Patrick Brennan of Juola Associates) consists of extracts from published textbooks on computer science and related disciplines, culled from an on-line repository. A pool of 16 authors was selected and their works were collected. Each test and training document was around 1,000 words. weekly newspaper TO BHMA2 from 1996 to 2012. The length of each article is at least 1,000 words.

	Training	Test
Genre	cases	cases
English	10	30
Greek	20	30
Spanish	10	30
$\sum$		

Table 43: CLEF-PAN 2013 Corpus Details

5.4 Experiments and Results

5.4.1 Data Collection

The corpus used consists of four publicly accessible CLEF-PAN shared tasks from writers who wrote in groups or pairs across various genres, languages and lengths as shown in Section 5.3. These include CLEF-PAN 2013, 2015, 2015, 2021. The classifiers created by the different approaches are configured using the training corpus defined during the CLEF-PAN campaigns, then we evaluated them using test sets also provided by CLEF-PAN.

5.4.2 Preprocessing

All the data sets underwent the same preprocessing procedure, just like with author profiling. Conversion to lowercase, text lemmatization, text stemming, all url changes to "urllink". In addition contraction have been changed to a two-word word (e.g., don't (do + not) becomes donot, cannot (can + not) becomes cannot). The sole distinction is that all generated tokens were taken into account rather than any words being eliminated from the text.

Class	fandoms
True	I shift a bit, warily letting my eyes dart from one owl to the other – but my eyes are trained on the Barn Owl the most. Like Hoole...so like Hoole... He turns a bit, and our eyes meet directly. I can"t describe it...in this next moment, I don"t look away, how awkward it seems. I stare into his eyes. They"re like Hoole"s... They are Barn Owl eyes, but Hoole"s eyes. They"re his eyes...Hoole"s eyes... They hold that light of valor, justice, that one glow that I always made me feel my gizzard twitch in the bottom of my heart. Hoole... All will become one with Russia," he said, almost simply, his cheer eerie. Fists were already clenched; now they groped about, for a pan, a rifle, a sword-there was nothing. In some way, this brought her but a sigh of relief-Gilbert and Roderich, she was reminded, were not here to suffer as well. If Ivan put his giant hands on Roderich...
False	Max," inquired Sam, "are you going to come tomorrow, too?" "Wouldn"t miss it," he responded. Then Jimmy called him over to his desk. "Walker, do you have that item I asked you for?" Sam queried. "Right here," he rejoined as he handed her a large manila envelope that was stiff. "I"m still curious about what you"re going to do with this." She just smiled and thanked him as she took it, then made her way toward Syd and Gage. Sydney"s crutches leaned against her desk as she sat in her chair. She quietly watched the activities in the room with a pleased smile. So, did I really fuck him?" Runa looked concerned now. Darina started laughing "What"s so funny?" "That you actually fell for that. Murphy isn"t the kind of guy that"ll have sex with someone that"s had more to drink then him, no matter how attractive she is" "What"s that supposed to mean? I thought that both of them want you" "Yeah...

Table 44: Sample of author verification text snippet pairs showing two classes

5.4.3 Feature Selection and Text Representation

In our experiments five feature sets have been used. First the entire vocabulary set is used. Then the term frequency set, which includes terms that occurs frequently in the documents sorted in descending order and taking the top 300 features. The third is a TFIDF feature set, which consists of tokens that are considered to be relevant for a document classification. The term frequency (TF) of a word in a document is multiplied by the inverse document frequency (IDF) of that word. This results into the TF-IDF score of a word in a document from which 300 features with the top scores are selected. The higher the score, the more relevant that word is in that particular document. The fourth is a χ^2 feature set described in Section 2.2.6, which consists of filtered set of words that have a high dependency with a given class, while those with low dependency are left out. Finally, the fifth is a PMI feature set described in Section 2.2.4, which consists of tokens extracted using PMI score which is the measure of association between a feature (a word) and a class or category. The performance of the models are then compared to the models built with all features. Where the features have a score such as TF, TF-IDF, χ^2 and PMI, we selected the top 300 features.

The text pairs or text sets are then represented each by the listed feature set values. These value will generate a vector for each text. A difference of the two vectors are obtained to give a vector

representation of the pairs (see Section 2.3). When one can observe a high similarity in the text pairs, the vector will have small differences on the other hand, the difference between vectors will be large. This difference vector representation will then be used in the different classifiers.

5.4.4 Classification

We used KNN, SVM, ExtraTrees, Decision Tree, Gaussian NB, Bernoulli NB, Multinomial NB, MLP, SGD, LDA, Random Forest, AdaBoost, Bagging, Gradient Boosting, XGB, and Logistic Regresion in accordance with the same configuration as in the scikit-learn library with the default parameter setting for all the experiments.

Testing the performance of the listed classifiers using five feature sets listed in Section 5.4.3 we obtain results for area under the curve (AUC), c@1 as well as accuracy values. We analyse the statistical significance difference in accuracies obtained across models used in the different datasets.

Each classifier is used to create five models, for 16 classifiers we have a total of 5*16 = 80 models for each dataset.

5.4.5 Overall Performance of the Models

Tables 45, 47, 49, 50, 52 present the comparison of the proposed approaches on 16 classifiers applied to 4 authorship verification datasets in different languages. Table 45 contains the evaluation in terms of final score (FS) which is a product of AUC and c@1 (AUC * c@1), Area under the curve (AUC), c@1 and accuracy for English, Greek, Spanish languages. Tables 47, Tables 48 and 49 contain the result of 2014 for different languages and text genres. They include English essay and novels, Spanish articles, Dutch essays and reviews, Greek articles. Table 50 contains the 2015 evaluation results in four different languages namely English, Greek, Dutch and Spanish. Table 52 contains results of 2021 including F1-score, AUC, c@1, Bier, overall values obtained from the mean of the four measures and accuracy.

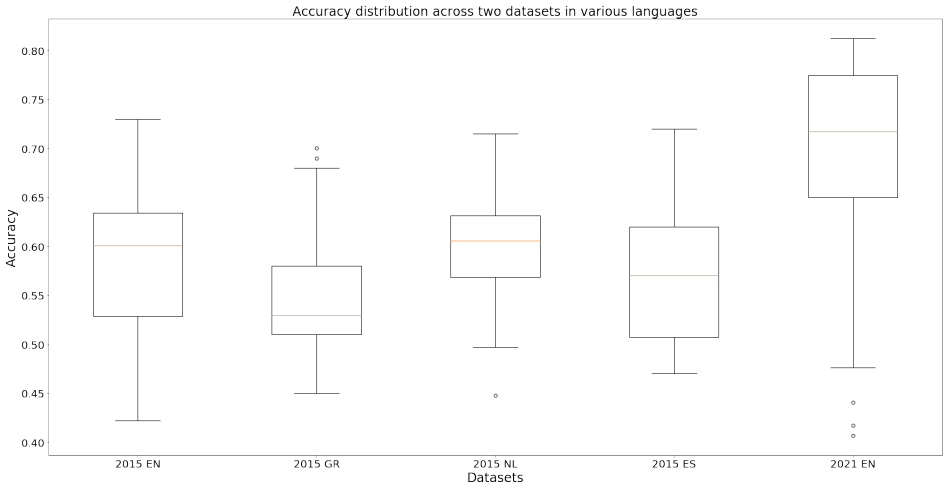


Figure 13: Accuracy distribution across datasets used in 2015 and 2021 in four languages

As for all evaluation measures, there is always some random aspects and usually a range of possible values make more sense. To achieve this, Figure 13 shows the accuracy distribution for the author verification task in two datasets across four languages. We will therefore talk about general performance in terms of the classifiers and the features in the various data.

Figure 13 also indicates that the mean performance across four languages is between 0.55 to 0.6 for 2015 and higher (0.72) for 2021 dataset. Even if we have the same number of problems for the four languages in 2015 (100 problems), the accuracy distribution is not the same. For Greek, the range of performance values are smaller than for the English collection

When using term frequency features, the bagging classifier produced the best accuracy of 0.733 with the English data in 2013 presented in Table 45. It is interesting to note that using all features does not always provide the best accuracy, even with learning schemes including a feature selection procedure such as decision trees or ExtraTrees. When all features were employed in the Spanish articles, Gradient Boosting classifiers yielded better accuracy for 2014 of 0.840 indicated in bold in Table 49. For the English dataset in 2015, logistic regression with TFIDF features produced the best accuracy of 0.730 shown in bold in Table 50. For the year 2021, SVM using TFIDF features produced the best accuracy of 0.812 (in bold in Table 52).

The performance differences between the best system and the following systems are not always statistically significant. For instance, as indicated in Table 52, the MLP with TF and TFIDF produced an accuracy rate of 0.807, while the best performing system was an SVM employing TFIDF features (0.812), an absolute difference of 0.047 (0.62%). Our statistical analysis determined that this difference is not significant. As another example Bagging classifier had the highest accuracy rate of 0.715, (similar to 2015 Dutch articles 2015 corpus (see Table 50), whereas Random Forest with PMI features gave a performance of 0.685. The performance difference is 0.03 (4.38%) was not statistically significant. To indicate statistically significant performance difference with the best system, we add an asterik (*) in all tables after the corresponding run.

Figure 14 displays models from Table 52 containing 2021 author verification accuracies that have significant differences (appear below the blue line) from the top model (SVM with TFIDF feature set) and those that do not (appear above the blue horizontal line).

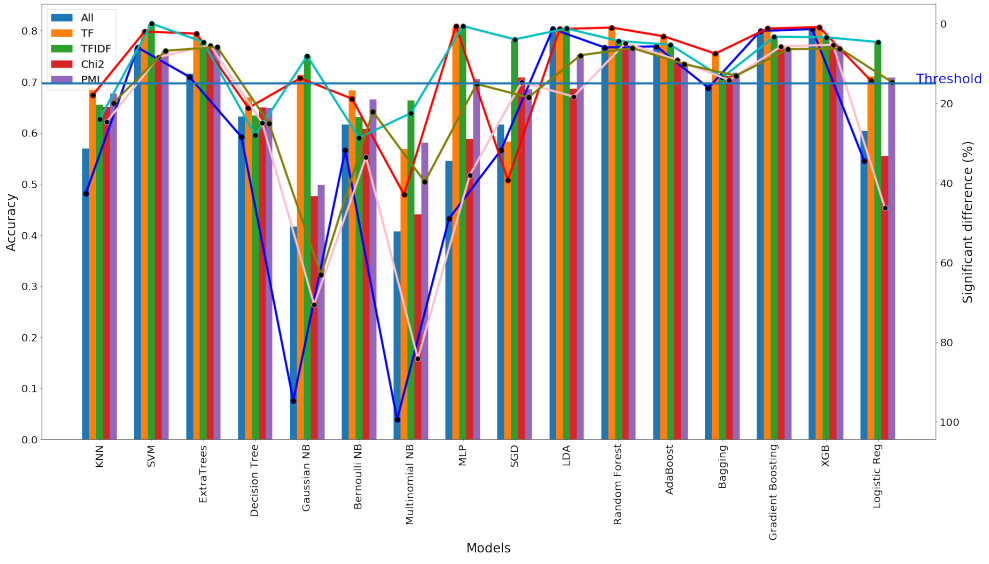


Figure 14: Statistically significant differences among the models in 2021 author verification

The overall effectiveness of the algorithms while using the various feature sets is displayed in the Table 53. For example Random Forest with with χ^2 and PMI features were the overall best performers with seven out of the 16 models getting the best score. These were followed by MLP with TFIDF features, Random forest with TF and TFIDF features, AdaBoost with all and χ^2 features, and XGB with TFIDF features. The best scoring models include TFIDF feature set using the top 300 features. The main advantage of TFIDF weighting scheme is that it considers both frequent terms and terms

seen present in few documents. When focusing on the style, our prior feeling was that TF would be the best approach. Our experiments show that this is second best one. Both χ^2 and PMI do not perform well, having the tendency to favour less frequently used terms as soon as they appear mainly in a single category. For the feature scoring functions TF, TFIDF, χ^2 and PMI only the top 300 features are selected to be used.

The data depicted in Table 54 shows that a strong correlation does exist between all performance measurements. In particular, the FS and AUC are usually strongly correlated than the other measurements, as well as the pair accuracy rate and c@1. In addition, one can observe that AUC and c@1 present usually a less strong correlation. This could be explained by the fact that c@1 is based only on the first result unlike the AUC. The smallest correlation value can be found between accuracy rate and AUC for the 2015 Spanish corpus. As a main conclusion, Table 54 clearly support the fact that the four performance measurements are correlated. Thus taking account of one measure is enough to draw conclusion about the effectiveness of different machines learning models.

English 2013

Model	Feature	FS	AUC	c@1	Acc
KNN	All	0.360	0.574	0.533	0.533*
	TF	0.353	0.623	0.567	0.567*
	TFIDF	0.265	0.567	0.467	0.467*
	Chi2	0.248	0.531	0.467	0.467*
	PMI	0.176	0.406	0.433	0.433*
SVM	All	0.304	0.536	0.567	0.500*
	TF	0.215	0.460	0.467	0.467*
	TFIDF	0.052	0.121	0.433	0.567*
	Chi2	0.308	0.527	0.586	0.433*
	PMI	0.154	0.344	0.448	0.467*
ExtraTrees	All	0.328	0.656	0.500	0.500*
	TF	0.428	0.676	o.633	0.633*
	TFIDF	0.522	0.797	0.654	0.633*
	Chi2	0.344	0.607	0.567	0.567*
	PMI	0.360	0.636	0.567	0.533*
Decision Tree	All	0.446	0.670	0.667	0.667
	TF	0.446	0.670	0.667	0.667
	TFIDF	0.212	0.455	0.433	0.433*
	Chi2	0.281	0.527	0.533	0.533*
	PMI	0.446	0.670	0.667	0.667
Gaussian NB	All	0.316	0.558	0.567	0.567*
	TF	0.340	0.638	0.533	0.533*
	TFIDF	0.222	0.513	0.433	0.433*
	Chi2	0.286	0.536	0.533	0.533*
	PMI	0.351	0.585	0.600	0.600*
Bernoulli NB	All	0.218	0.545	0.400	0.400*
	TF	0.383	0.638	0.600	0.600 *
	TFIDF	0.269	0.504	0.533	0.533*
	Chi2	0.146	0.366	0.400	0.400*
	PMI	0.120	0.328	0.367	0.367*
Multi. NB	All	0.228	0.455	0.500	0.500*
	TF	0.288	0.509	0.567	0.567*
	TFIDF	0.400	0.750	0.533	0.533*
	Chi2	0.228	0.455	0.500	0.500*
	PMI	0.163	0.406	0.400	0.400*
MLP	All	0.244	0.522	0.467	0.467*
	TF	0.367	0.612	0.600	0.600*
	TFIDF	0.074	0.701	0.533	0.533*
	Chi2	0.240	0.513	0.467	0.467*
	PMI	0.269	0.504	0.533	0.533*
SGD	All	0.238	0.446	0.533	0.533*
	TF	0.237	0.509	0.467	0.467*
	TFIDF	0.41	0.830	0.567	0.567*
	Chi2	0.208	0.446	0.467	0.467*
	PMI	0.214	0.402	0.533	0.533*
LDA	All	0.300	0.562	0.533	0.533*
	TF	0.362	0.638	0.567	0.567*
	TFIDF	0.271	0.509	0.533	0.533 *
	Chi2	0.362	0.638	0.567	0.567*
	PMI	0.383	0.638	0.600	0.600*
Random Forest	All	0.449	0.721	0.623	0.667
	TF	0.319	0.598	0.533	0.567*
	TFIDF	0.395	0.634	0.623	0.600*
	Chi2	0.394	0.692	0.569	0.567*
	PMI	0.248	0.531	0.467	0.467*
AdaBoost	All	0.281	0.527	0.533	0.533*
	TF	0.446	0.670	0.667	0.667
	TFIDF	0.212	0.455	0.467	0.467*
	Chi2	0.217	0.464	0.467	0.467*
	PMI	0.331	0.585	0.567	0.567*
Bagging	All	0.276	0.531	0.520	0.567*
	TF	0.584	0.792	0.737	**0.733**
	TFIDF	0.234	0.462	0.506	0.533*
	Chi2	0.489	0.763	0.640	0.633*
	PMI	0.446	0.717	0.622	0.633*
Gradient Boosting	All	0.384	0.641	0.600	0.600*
	TF	0.442	0.699	0.633	0.633*
	TFIDF	0.212	0.455	0.467	0.467*
	Chi2	0.469	0.703	0.667	0.667
	PMI	0.382	0.636	0.600	0.600*
XGB	All	0.321	0.567	0.567	0.567*
	TF	0.321	0.567	0.567	0.567*
	TFIDF	0.212	0.455	0.467	0.467*
	Chi2	0.321	0.567	0.567	0.567*
	PMI	0.321	0.567	0.567	0.567*
Logistic Reg	All	0.228	0.455	0.500	0.500*
	TF	0.266	0.531	0.500	0.500*
	TFIDF	0.460	0.862	0.533	0.533*
	Chi2	0.234	0.469	0.500	0.500*
	PMI	0.163	0.406	0.400	0.400*

Greek 2013

Model	Feature	FS	AUC	c@1	Acc
KNN	All	0.215	0.460	0.467	0.467*
	TF	0.204	0.471	0.433	0.433*
	TFIDF	0.169	0.422	0.400	0.400*
	Chi2	0.233	0.411	0.567	0.567*
	PMI	0.349	0.551	0.633	0.633
SVM	All	0.136	0.347	0.391	0.533*
	TF	0.143	0.378	0.378	0.500*
	TFIDF	0.143	0.402	0.356	0.567*
	Chi2	0.154	0.480	0.320	0.533*
	PMI	0.301	0.564	0.533	0.567*
ExtraTrees	All	0.446	0.638	0.700	**0.700**
	TF	0.283	0.613	0.462	0.467*
	TFIDF	0.484	0.740	0.654	0.667
	Chi2	0.392	0.629	0.623	0.600*
	PMI	0.363	0.638	0.569	0.567*
Decision Tree	All	0.360	0.600	0.600	0.600*
	TF	0.188	0.433	0.433	0.433*
	TFIDF	0.134	0.367	0.367	0.367*
	Chi2	0.321	0.567	0.567	0.567*
	PMI	0.360	0.600	0.600	0.600 *
Gaussian NB	All	0.188	0.433	0.433	0.433*
	TF	0.277	0.520	0.533	0.533*
	TFIDF	0.349	0.551	0.633	0.633
	Chi2	0.284	0.533	0.533	0.533*
	PMI	0.321	0.567	0.567	0.567*
Bernoulli NB	All	0.302	0.533	0.567	0.567*
	TF	0.352	0.587	0.600	0.600*
	TFIDF	0.427	0.640	0.667	0.667
	Chi2	0.325	0.573	0.567	0.567*
	PMI	0.264	0.529	0.500	0.500*
Multi. NB	All	0.342	0.684	0.500	0.500*
	TF	0.369	0.738	0.500	0.500*
	TFIDF	0.325	0.542	0.600	0.600*
	Chi2	0.333	0.667	0.500	0.500*
	PMI	0.187	0.373	0.500	0.500*
MLP	All	0.544	0.778	0.700	0.700*
	TF	0.391	0.782	0.500	0.500*
	TFIDF	0.294	0.551	0.533	0.533*
	Chi2	0.439	0.693	0.693	0.693
	PMI	0.342	0.684	0.500	0.500*
SGD	All	0.349	0.698	0.500	0.500*
	TF	0.376	0.751	0.500	0.500*
	TFIDF	0.245	0.524	0.467	0.467*
	Chi2	0.331	0.662	0.500	0.500*
	PMI	0.182	0.364	0.500	0.500*
LDA	All	0.160	0.436	0.367	0.367*
	TF	0.340	0.600	0.567	0.567*
	TFIDF	0.358	0.631	0.567	0.567*
	Chi2	0.332	0.622	0.533	0.533*
	PMI	0.235	0.440	0.533	0.533*
Random Forest	All	0.398	0.680	0.586	0.567*
	TF	0.388	0.582	0.667	0.667
	TFIDF	0.395	0.653	0.604	0.633
	Chi2	0.370	0.631	0.586	0.600*
	PMI	0.468	0.680	0.689	**0.700**
AdaBoost	All	0.263	0.527	0.500	0.500*
	TF	0.185	0.427	0.433	0.433*
	TFIDF	0.167	0.418	0.400	0.400*
0.381	0.636	0.600	0.600*		
	PMI	0.417	0.696	0.600	0.600*
Bagging	All	0.239	0.484	0.493	0.533*
	TF	0.436	0.700	0.622	**0.700**
	TFIDF	0.222	0.500	0.444	0.467*
	Chi2	0.364	0.624	0.583	0.600*
	PMI	0.413	0.689	0.600	0.600*
Gradient Boosting	All	0.376	0.627	0.600	0.600*
	TF	0.246	0.527	0.467	0.467*
	TFIDF	0.199	0.427	0.467	0.467*
	Chi2	0.373	0.622	0.600	0.600*
	PMI	0.389	0.649	0.600	0.600*
XGB	All	0.387	0.644	0.600	0.600*
	TF	0.270	0.507	0.533	0.533*
	TFIDF	0.292	0.547	0.533	0.533*
	Chi2	0.355	0.591	0.600	0.600*
	PMI	0.373	0.622	0.600	0.600*
Logistic Reg	All	0.453	0.680	0.667	0.667
	TF	0.476	0.738	0.633	0.633
	TFIDF	0.341	0.569	0.600	0.600*
	Chi2	0.436	0.653	0.667	0.667
	PMI	0.187	0.373	0.500	0.500*

Table 45: PAN 2013 Author verification results part 1.

Spanish 2013

Model	Feature	FS	AUC	c@1	Acc
KNN	All	0.260	0.500	0.520	0.520*
	TF	0.260	0.500	0.520	0.520*
	TFIDF	0.260	0.500	0.520	0.520*
	Chi2	0.260	0.500	0.520	0.520*
	PMI	0.260	0.500	0.520	0.520*
SVM	All	0.067	0.231	0.291	0.520*
	TF	0.160	0.321	0.499	0.520*
	TFIDF	0.118	0.295	0.400	0.520*
	Chi2	0.063	0.224	0.280	0.520*
	PMI	0.439	0.686	0.640	0.600
ExtraTrees	All	0.379	0.651	0.582	0.600
	TF	0.328	0.587	0.560	0.560*
	TFIDF	0.270	0.481	0.562	0.560*
	Chi2	0.365	0.702	0.520	0.520*
	PMI	0.300	0.497	0.605	0.560*
Decision Tree	All	0.406	0.635	0.640	0.640
	TF	0.190	0.433	0.440	0.440*
	TFIDF	0.272	0.522	0.520	0.520*
	Chi2	0.226	0.471	0.480	0.480*
	PMI	0.460	0.676	0.680	**0.680**
Gaussian NB	All	0.260	0.500	0.520	0.520*
	TF	0.260	0.500	0.520	0.520*
	TFIDF	0.260	0.500	0.520	0.520*
	Chi2	0.228	0.474	0.480	0.480*
	PMI	0.260	0.500	0.520	0.520*
Bernoulli NB	All	0.297	0.571	0.520	0.520*
	TF	0.262	0.503	0.520	0.520*
	TFIDF	0.277	0.532	0.520	0.520*
	Chi2	0.260	0.500	0.520	0.520*
	PMI	0.221	0.503	0.440	0.440*
Multi NB	All	0.310	0.596	0.520	0.520*
	TF	0.290	0.558	0.520	0.520*
	TFIDF	0.183	0.353	0.520	0.520*
	Chi2	0.317	0.609	0.520	0.520*
	PMI	0.27	0.628	0.520	0.520*
MLP	All	0.373	0.718	0.520	0.520*
	TF	0.407	0.782	0.520	0.520*
	TFIDF	0.240	0.460	0.520	0.520*
	Chi2	0.353	0.679	0.520	0.520*
	PMI	0.343	0 660	0 520	0.520*
SGD	All	0.380	0.731	0.520	0.520*
	TF	0.387	0.744	0.520	0.520*
	TFIDF	0.357	0.686	0.520	0.520*
	Chi2	0.395	0.705	0.560	0.560*
	PMI	0.305	0.635	0.480	0.480*
LDA	All	0.230	0.442	0.520	0.520*
	TF	0.277	0.532	0.520	0.520*
	TFIDF	0.267	0.513	0.520	0.520*
	Chi2	0.250	0.481	0.520	0.520*
	PMI	0.243	0.506	0.480	0.480*
Random Forest	All	0.255	0.471	0.541	0.560*
	TF	0.312	0.599	0.520	0.520*
	TFIDF	0.266	0.474	0.560	0.560*
	Chi2	0.372	0.715	0.520	0.520*
	PMI	0.228	0.439	0.518	0.520*
AdaBoost	All	0.406	0.635	0.640	0.64
	TF	0.458	0.673	0.680	0.680
	TFIDF	0.132	0.365	0.360	0.360*
	Chi2	0.260	0.500	0.520	0.520*
	PMI	0.305	0.545	0.560	0.560*
Bagging	All	0.279	0.519	0.536	0.560*
	TF	0.189	0.407	0.464	0.480*
	TFIDF	0.335	0.619	0.541	0.560 *
	Chi2	0.284	0.506	0.562	0.600
	PMI	0.303	0.526	0.576	0.640
Gradient Boosting	All	0.316	0.564	0.560	0.560*
	TF	0.412	0.686	0.600	0.600
	TFIDF	0.287	0.513	0.560	0.560*
	Chi2	0.277	0.532	0.520	0.520*
	PMI	0.316	0.564	0.560	0.560*
XGB	All	0.260	0.500	0.520	0.520*
	TF	0.260	0.500	0.520	0.520*
	TFIDF	0.260	0.500	0.520	0.520*
	Chi2	0.260	0.500	0.520	0.520*
	PMI	0.260	0.500	0.520	0.520*
Logistic Reg	All	0.363	0.699	0.520	0.520*
	TF	0.373	0.718	0.520	0.520*
	TFIDF	0.357	0.686	0.520	0.520*
	Chi2	0.370	0.712	0.520	0.520*
	PMI	0.330	0.635	0.520	0.520*

Overall 2013

Model	Feature	FS	AUC	c@1	Acc
KNN	All	0.342	0.594	0.576	0.576*
	TF	0.313	0.566	0.553	0.553*
	TFIDF	0.268	0.517	0.518	0.518*
	Chi2	0.294	0.520	0.565	0.565*
	PMI	0.289	0.522	0.553	0.553*
SVM	All	0.205	0.458	0.447	0.541*
	TF	0.378	0.648	0.583	0.576*
	TFIDF	0.101	0.289	0.349	0.576*
	Chi2	0.235	0.498	0.471	0.506*
	PMI	0.225	0.508	0.443	0.600*
Decision Tree	All	0.362	0.593	0.610	0.612
	TF	0.378	0.648	0.583	0.576*
	TFIDF	0.470	0.699	0.673	0.671
	Chi2	0.400	0.657	0.609	0.612*
	PMI	0.356	0.637	0.559	0.565*
ExtraTrees	All	0.347	0.589	0.588	0.588*
	TF	0.243	0.492	0.494	0.494*
	TFIDF	0.305	0.552	0.553	0.553*
	Chi2	0.293	0.541	0.541	0.541*
	PMI	0.360	0.600	0.600	0.600*
Gaussian NB	All	0.211	0.460	0.459	0.459*
	TF	0 315	0.569	0.553	0.553*
	TFIDF	0.323	0.584	0.553	0.553*
	Chi2	0.203	0.443	0.459	0.459*
	PMI	0.265	0.512	0.518	0.518*
Bernoulli NB	All	0.288	0.533	0.541	0.541*
	TF	0.254	0.514	0.494	0.494*
	TFIDF	0.261	0.527	0.494	0.494*
	Chi2	0.249	0.504	494	0.494*
	PMI	0.312	0.590	0.529	0.529*
Multi NB	All	0.259	0.523	0.494	0.494*
	TF	0.263	0.533	0.494	0.494*
	TFIDF	0.313	0.543	0.576	0.576*
	Chi2	0.269	0.545	0.494	0.494*
	PMI	0.251	0.508	0.494	0.494*
MLP	All	0.333	0.590	0.565	0.565*
	TF	0.283	0.572	0.494	0.494*
	TFIDF	0.362	0.603	0.600	0.600
	Chi2	0.305	0.576	0.529	0.529*
	PMI	0.283	0.572	0.494	0.494*
SGD	All	0.279	0.551	0.506	0.506*
	TF	0.276	0.559	0.494	0.494*
	TFIDF	0.377	0.642	0.588	0.588
	Chi2	0.287	0.581	0.494	0.494*
	PMI	0.258	0.509	0.506	0.506*
LDA	All	0.300	0.543	0.553	0.553*
	TF	0.249	0 493	0.506	0.506*
	TFIDF	0.274	0.517	0.529	0.529*
	Chi2	0.360	0.612	0.588	0.588
	PMI	0.180	0.477	0.376	0.376*
Random Forest	All	0.299	0.585	0.512	0.529*
	TF	0.439	0.683	0.643	0.643
	TFIDF	0.353	0.623	0.567	0.576*
	Chi2	0.376	0.612	0.614	0.612
	PMI	0.395	0.643	0.614	0.612
AdaBoost	All	0.433	0.682	0.635	0.635
	TF	0.398	0.650	0.612	0.612
	TFIDF	0.302	0.558	0.541	0.541*
	Chi2	0.439	0.704	0.624	0.624
	PMI	0.307	0.568	0.541	0.541*
Bagging	All	0.482	0.732	0.658	**0.671**
	TF	0.337	0.604	0.559	0.541*
	TFIDF	0.299	0.554	0.540	0.518*
	Chi2	0.335	0.584	0.572	0.588
	PMI	0.373	0.624	0.597	0.600
Gradient Boosting	All	0.332	0.587	0.565	0.565*
	TF	0.398	0.651	0.612	0.612
	TFIDF	0.341	0.569	0.600	0.600
	Chi2	0.345	0.564	0.612	0.612
	PMI	0.381	0.634	0.600	0.600
XGB	All	0.355	0.642	0.553	0.553*
	TF	0.306	0.542	0.565	0.565*
	TFIDF	0.382	0.637	0.600	0.600
	Chi2	0.390	0.650	0.600	0.600
	PMI	0.333	0.629	0.529	0.529*
Logistic Reg	All	0.270	0.546	0.494	0.494*
	TF	0.273	0.553	0.494	0.494*
	TFIDF	0.326	0.615	0.529	0.529*
	Chi2	0.274	0.554	0.494	0.494*
	PMI	0.252	0.510	0.494	0.494*

Table 46: PAN 2013 Author verification results part 2.

English essays 2014

Model	Feature	FS	AUC	c@1	Acc
KNN	All	0.298	0.551	0.540	0.540*
	TF	0.372	0.620	0.600	0.600*
	TFIDF	0.313	0.579	0.540	0.540*
	Chi2	0.294	0.520	0.565	0.565*
	PMI	0.350	0.583	0.600	0.600*
SVM	All	0.403	0.655	0.616	0.620*
	TF	0.434	0.693	0.625	0.590*
	TFIDF	0.540	0.760	0.710	0.670
	Chi2	0.240	0.498	0.481	0.506*
	PMI	0.296	0.560	0.529	0.553
ExtraTrees	All	0.330	0.587	0.562	0.580*
	TF	0.430	0.675	0.637	0.630*
	TFIDF	0.463	0.719	0.645	0.620*
	Chi2	0.364	0.595	0.612	0.612*
	PMI	0.356	0.608	0.585	0.585
Decision Tree	All	0.230	0.480	0.480	0.480*
	TF	0.423	0.650	0.650	0.650
	TFIDF	0.325	0.570	0.570	0.570*
	Chi2	0.346	0.589	0.588	0.588*
	PMI	0.211	0.459	0.459	0.459*
Gaussian NB	All	0.144	0.380	0.520	0.520*
	TF	0.447	0.698	0.640	0.640
	TFIDF	0.494	0.727	0.680	0.680
	Chi2	0.203	0.443	0.459	0.459*
	PMI	0.251	0.496	0.506	0.506*
Bernoulli NB	All	0.253	0.486	0.520	0.520*
	TF	0.399	0.665	0.600	0.600*
	TFIDF	0.336	0.589	0.570	0.570*
	Chi2	0.249	0.504	0.494	0.494*
	PMI	0.285	0.551	0.518	0.518*
Multi. NB	All	0.376	0.659	0.570	0.570*
	TF	0.362	0.695	0.520	0.520*
	TFIDF	0.529	0.778	0.680	0.680
	Chi2	0.269	0.545	0.494	0.494*
	PMI	0.258	0.522	0.494	0.494*
MLP	All All	0.438	0.707	0.620	0.620*
	TF	0.345	0.690	0.500	0.500*
	TFIDF	0.509	0.749	0.680	0.680
	Chi2	0.296	0.572	0.518	0.518*
	PMI	0.280	0.568	0.494	0.494*
SGD	All	0.314	0.628	0.500	0.500*
	TF	0.340	0.680	0.500	0.500*
	TFIDF	0.474	0.729	0.650	0.650
	Chi2	0.286	0.565	0.506	0.506*
	PMI	0.265	0.535	0.494	0.494*
LDA	All	0.178	0.396	0.450	0.450*
	TF	0.388	0.626	0.620	0.620*
	TFIDF	0.416	0.682	0.610	0.610*
	Chi2	0.360	0.612	0.588	0.588*
	PMI	0.235	0.487	0.482	0.482*
Random Forest	All	0.378	0.660	0.572	0.580*
	TF	0.424	0.686	0.618	0.620*
	TFIDF	0.521	0.767	0.678	0.660
	Chi2	0.337	0.631	0.535	0.529*
	PMI	0.385	0.615	0.626	0.635
AdaBoost	All	0.476	0.690	0.690	0.690
	TF	0.520	0.743	0.700	0.700
	TFIDF	0.417	0.662	0.630	0.630*
	Chi2	0.423	0.692	0.612	0.612*
	PMI	0.334	0.604	0.553	0.553*
Bagging	All	0.223	0.476	0.468	0.450*
	TF	0.334	0.591	0.566	0.580*
	TFIDF	0.343	0.596	0.575	0.620*
	Chi2	0.289	0.558	0.518	0.541*
	PMI	0.327	0.594	0.551	0.565*
Gradient Boosting	All	0.274	0.538	0.510	0.510*
	TF	0.402	0.648	0.620	0.620*
	TFIDF	0.524	0.770	0.680	0.680
	Chi2	0.299	0.578	0.518	0.518*
	PMI	0.336	0.594	0.565	0.565*
XGB	All	0.380	0.644	0.590	0.590*
	TF	0.378	0.630	0.600	0.600*
	TFIDF	0.437	0.672	0.650	0.650
	Chi2	0.390	0.650	0.600	0.600*
	PMI	0.316	0.559	0.565	0.565*
Logistic Reg	All	0.344	0.625	0.550	0.550*
	TF	0.416	0.672	0.620	0.620*
	TFIDF	0.579	0.793	0.730	**0.730**
	Chi2	0.274	0.554	0.494	0.494*
	PMI	0.258	0.522	0.494	0.494*

English novels 2014

Model	Feature	FS	AUC	c@1	Acc
KNN	All	0.297	0.619	0.480	0.480*
	TF	0.358	0.651	0.550	0.550*
	TFIDF	0.377	0.650	0.580	0.580*
	Chi2	0.302	0.629	0.480	0.480*
	PMI	0.275	0.560	0.490	0.490*
SVM	All	0.430	0.682	0.630	0.620*
	TF	0.448	0.700	0.640	0.630
	TFIDF	0.461	0.710	0.650	0.660
	Chi2	0.402	0.653	0.616	0.600*
	PMI	0.350	0.603	0.580	0.560*
ExtraTrees	All	0.522	0.763	0.683	0.680
	TF	0.541	0.804	0.673	0.670
	TFIDF	0.461	0.736	0.626	0.630
	Chi2	0.578	0.813	0.711	0.710
	PMI	0.472	0.721	0.655	0.640
Decision Tree	All	0.221	0.470	0.470	0.470*
	TF	0.176	0.420	0.420	0.420*
	TFIDF	0.281	0.530	0.530	0.530*
	Chi2	0.168	0.410	0.410	0.410*
	PMI	0.194	0.440	0.440	0.440*
Gaussian NB	All	0.281	0.530	0.530	0.530*
	TF	0.389	0.628	0.620	0.620*
	TFIDF	0.451	0.683	0.660	0.660
	Chi2	0.351	0.595	0.590	0.590*
	PMI	0.258	0.506	0.510	0.510*
Bernoulli NB	All	0.357	0.605	0.590	0.590*
	TF	0.432	0.720	0.600	0.600*
	TFIDF	0.425	0.664	0.640	0.640
	Chi2	0.502	0.772	0.650	0.650
	PMI	0.388	0.636	0.610	0.610*
Multi. NB	All	0.364	0.728	0.500	0.500*
	TF	0.366	0.731	0.500	0.500*
	TFIDF	0.345	0.677	0.510	0.510*
	Chi2	0.360	0.721	0.500	0.500*
	PMI	0.344	0.689	0.500	0.500*
MLP	All	0.484	0.756	0.640	0.640
	TF	0.480	0.716	0.670	0.670
	TFIDF	0.481	0.697	0.690	0.690
	Chi2	0.455	0.734	0.620	0.620*
	PMI	0.347	0.694	0.500	0.500*
SGD	All	0.365	0.730	0.500	0.500*
	TF	0.368	0.736	0.500	0.500*
	TFIDF	0.398	0.687	0.580	0.580*
	Chi2	0.363	0.726	0.500	0.500*
	PMI	0.345	0.690	0.500	0.500
LDA	All	0.476	0.807	0.590	0.590*
	TF	0.453	0.768	0.590	0.590*
	TFIDF	0.377	0.685	0.550	0.550*
	Chi2	0.358	0.639	0.560	0.560*
	PMI	0.280	0.549	0.510	0.510
Random Forest	All	0.583	0.826	0.706	**0.720**
	TF	0.474	0.738	0.643	0.640
	TFIDF	0.415	0.688	0.603	0.600*
	Chi2	0.562	0.794	0.707	0.710
	PMI	0.499	0.745	0.670	0.660
AdaBoost	All	0.352	0.608	0.580	0.580*
	TF	0.370	0.637	0.580	0.580 *
	TFIDF	0.379	0.643	0.590	0.590*
	Chi2	0.453	0.708	0.640	0.640
	PMI	0.446	0.698	0.640	0.640
Bagging	All	0.298	0.563	0.529	0.530*
	TF	0.328	0.589	0.557	0.520*
	TFIDF	0.253	0.508	0.499	0.510*
	Chi2	0.200	0.467	0.428	0.440*
	PMI	0.230	0.496	0.464	0.470*
Gradient Boosting	All	0.192	0.408	0.470	0.470*
	TF	0.222	0.504	0.440	0.440*
	TFIDF	0.260	0.530	0.490	0.490*
	Chi2	0.216	0.503	0.430	0.430 *
	PMI	0.257	0.547	0.470	0.470*
XGB	All	0.360	0.620	0.580	0.580*
	TF	0.307	0.558	0.550	0.550*
	TFIDF	0.309	0.551	0.560	0.560*
	Chi2	0.399	0.644	0.620	0.620*
	PMI	0.376	0.627	0.600	0.600*
Logistic Reg	All	0.451	0.728	0.620	0.620*
	TF	0.505	0.732	0.690	0.690
	TFIDF	0.448	0.689	0.650	0.650
	Chi2	0.451	0.727	0.620	0.620*
	PMI	0.428	0.690	0.620	0.620*

Table 47: PAN 2014 Author verification results: part 1.

Spanish articles 2014

Model	Feature	FS	AUC	c@1	Acc
KNN	All	0.491	0.743	0.660	0.660*
	TF	0.446	0.743	0.600	0.600*
	TFIDF	0.428	0.690	0.620	0.620*
	Chi2	0.513	0.778	0.660	0.660*
	PMI	0.299	0.554	0.540	0.540*
SVM	All	0.597	0.837	0.714	0.720*
	TF	0.719	0.899	0.800	0.780
	TFIDF	0.603	0.805	0.749	0.780
	Chi2	0.596	0.827	0.720	0.740
	PMI	0.473	0.717	0.660	0.680*
ExtraTrees	All	0.271	0.532	0.510	0.500*
	TF	0.738	0.910	0.811	0.800
	TFIDF	0.432	0.650	0.666	0.660
	Chi2	0.741	0.886	0.836	0.820
	PMI	0.381	0.634	0.600	0.600*
Decision Tree	All	0.360	0.600	0.600	0.600*
	TF	0.462	0.680	0.680	0.680*
	TFIDF	0.548	0.740	0.740	0.740
	Chi2	0.336	0.580	0.580	0.580*
	PMI	0.212	0.460	0.460	0.460*
Gaussian NB	All	0.250	0.500	0.500	0.500*
	TF	0.509	0.748	0.680	0.680*
	TFIDF	0.314	0.560	0.560	0.560*
	Chi2	0.386	0.644	0.600	0.600*
	PMI	0.168	0.400	0.420	0.420*
Bernoulli NB	All	0.219	0.457	0.480	0.480*
	TF	0.285	0.549	0.520	0.520*
	TFIDF	0.300	0.555	0.540	0.540*
	Chi2	0.320	0.616	0.520	0.520*
	PMI	0.237	0.474	0.500	0.500*
Multi. NB	All	0.357	0.714	0.500	0.500*
	TF	0.368	0.736	0.500	0.500*
	TFIDF	0.467	0.754	0.620	0.620*
	Chi2	0.358	0.717	0.500	0.500*
	PMI	0.354	0.707	0.500	0.500*
MLP	All	0.508	0.747	0.680	0.680*
	TF	0.370	0.741	0.500	0.500*
	TFIDF	0.457	0.693	0.660	0.660*
	Chi2	0.377	0.754	0.500	0.500*
	PMI	0.351	0.702	0.500	0.500*
SGD	All	0.354	0.707	0.500	0.500*
	TF	0.374	0.749	0.500	0.500*
	TFIDF	0.568	0.811	0.700	0.700*
	Chi2	0.364	0.728	0.500	0.500*
	PMI	0.354	0.707	0.500	0.500*
LDA	All	0.160	0.400	0.400	0.400*
	TF	0.408	0.618	0.660	0.660*
	TFIDF	0.238	0.475	0.500	0.500*
	Chi2	0.523	0.747	0.700	0.700*
	PMI	0.247	0.514	0.480	0.480*
Random Forest	All	0.570	0.762	0.748	0.760
	TF	0.610	0.847	0.720	0.720 *
	TFIDF	0.483	0.746	0.648	0.660*
	Chi2	0.609	0.821	0.742	0.720*
	PMI	0.280	0.538	0.520	0.520*
AdaBoost	All	0.422	0.680	0.620	0.620*
	TF	0.532	0.760	0.700	0.700*
	TFIDF	0.476	0.768	0.620	0.620*
	Chi2	0.430	0.693	0.620	0.620*
	PMI	0.287	0.574	0.500	0.500*
Bagging	All	0.383	0.624	0.614	0.600*
	TF	0.417	0.692	0.603	0.580*
	TFIDF	0.451	0.694	0.650	0.620*
	Chi2	0.576	0.776	0.742	0.720
	PMI	0.322	0.588	0.547	0.540*
Gradient Boosting	All	0.714	0.850	0.840	**0.840**
	TF	0.644	0.870	0.740	0.740
	TFIDF	0.530	0.757	0.700	0.700*
	Chi2	0.705	0.882	0.800	0.800
	PMI	0.334	0.619	0.540	0.540*
XGB	All	0.584	0.789	0.740	0.740
	TF	0.714	0.870	0.820	0.820
	TFIDF	0.482	0.730	0.660	0.660*
	Chi2	0.697	0.850	0.820	0.820
	PMI	0.313	0.579	0.540	0.540*
Logistic Reg	All	0.44	0.710	0.620	0.620*
	TF	0.520	0.742	0.700	0.700*
	TFIDF	0.551	0.787	0.700	0.700*
	Chi2	0.446	0.720	0.620	0.620*
	PMI	0.467	0.707	0.660	0.660*

Greek articles 2014

Model	Feature	FS	AUC	c@1	Acc
KNN	All	0.250	0.499	0.500	0.500*
	TF	0.209	0.454	0.460	0.460*
	TFIDF	0.568	0.789	0.720	**0.720**
	Chi2	0.093	0310	0.300	0.300*
	PMI	0.282	0.564	0.500	0.500*
SVM	All	0.253	0.496	0.510	0.500*
	TF	0.361	0.598	0.603	0.580*
	TFIDF	0.380	0.634	0.600	0.600*
	Chi2	0.217	0.451	0.480	0.480*
	PMI	0.404	0.632	0.640	0.600*
ExtraTrees	All	0.288	0.554	0.520	0.520*
	TF	0.483	0.712	0.678	0.680
	TFIDF	0.495	0.739	0.670	0.680
	Chi2	0.391	0.671	0.580	0.580*
	PMI	0.306	0.545	0.562	0.580*
Decision Tree	All	0.384	0.620	0.620	0.620*
	TF	0.250	0.500	0.500	0.500*
	TFIDF	0.360	0.600	0.600	0.600*
	Chi2	0.270	0.520	0.520	0.520*
	PMI	0.194	0.440	0.440	0.440*
Gaussian NB	All	0.336	0.580	0.580	0.580*
	TF	0.302	0.558	0.540	0.540*
	TFIDF	0.387	0.624	0.620	0.620*
	Chi2	0.343	0.571	0.600	0.600*
	PMI	0.253	0.486	0.520	0.520*
Bernoulli NB	All	0.186	0.422	0.440	0.440*
	TF	0.259	0.498	0.520	0.520*
	TFIDF	0.201	0.456	0.440	0.440*
	Chi2	0.207	0.470	0.440	0.440*
	PMI	0.236	0.514	0.460	0.460 *
Multinomial NB	All	0.202	0.374	0.540	0.540*
	TF	0.215	0.414	0.520	0.520*
	TFIDF	0.267	0.494	0.540	0.540*
	Chi2	0.186	0.358	0.520	0.520*
	PMI	0.284	0.568	0.500	0.500*
MLP	All	0.282	0.522	0.540	0.540*
	TF	0.366	0.731	0.500	0.500*
	TFIDF	0.283	0.566	0.500	0.500*
	Chi2	0.202	0.403	0.500	0.500*
	PMI	0.323	0.646	0.500	0.500*
SGD	All	0.194	0.387	0.500	0.500*
	TF	0.196	0.392	0.500	0.500*
	TFIDF	0.374	0.645	0.580	0.580*
	Chi2	0.174	0.349	0.500	0.500*
	PMI	0.282	0.565	0.500	0.500*
LDA	All	0.452	0.685	0.660	0.660
	TF	0.404	0.651	0.620	0.620*
	TFIDF	0.504	0.720	0.700	0.700
	Chi2	0.394	0.597	0.660	0.660
	PMI	0.435	0.659	0.660	0.660
Random Forest	All	0.252	0.520	0.484	0.500*
	TF	0.411	0.667	0.616	0.620*
	TFIDF	0.4832	0.690	0.700	0.700
	Chi2	0.315	0.583	0.540	0.520*
	PMI	0.272	0.580	0.469	0.480*
AdaBoost	All	0.357	0.595	0.600	0.600*
	TF	0.341	0.587	0.580	0.580*
	TFIDF	0.352	0.587	0.600	0.600*
	Chi2	0.306	0.589	0.520	0.520*
	PMI	0.258	0.517	0.500	0.500*
Bagging	All	0.316	0.562	0.561	0.580*
	TF	0.215	0.472	0.456	0.520*
	TFIDF	0.290	0.528	0.550	0.520*
	Chi2	0.356	0.605	0.589	0.600*
	PMI	0.241	0.495	0.487	0.540*
Gradient Boosting	All	0.375	0.626	0.600	0.600*
	TF	0.363	0.626	0.580	0.580*
	TFIDF	0.310	0.595	0.520	0.520*
	Chi2	0.340	0.586	0.580	0.580*
	PMI	0.222	0.506	0.440	0.440*
XGB	All	0.237	0.494	0.480	0.480*
	TF	0.256	0.533	0.480	0.480*
	TFIDF	0.489	0.718	0.680	0.680
	Chi2	0.350	0.603	0.580	0.580*
	PMI	0.262	0.525	0.500	0.500*
Logistic Reg	All	0.172	0.374	0.460	0.460*
	TF	0.171	0.406	0.420	0.420*
	TFIDF	0.405	0.675	0.600	0.600*
	Chi2	0.144	0.344	0.420	0.420*
	PMI	0.284	0.568	0.500	0.500*

Table 48: PAN 2014 Author verification results: part 2.

Dutch essays 2014

Model	Feature	FS	AUC	c@1	Acc
KNN	All	0.359	0.663	0.542	0.542*
	TF	0.320	0.615	0.521	0.521*
	TFIDF	0.351	0.648	0.542	0.542*
	Chi2	0.393	0.698	0.562	0.562*
	PMI	0.436	0.674	0.646	0.646*
SVM	All	0.429	0.710	0.604	0.604*
	TF	0.385	0.660	0.583	0.625*
	TFIDF	0.591	0.818	0.723	0.688
	Chi2	0.492	0.738	0.667	0.646*
	PMI 0.258	0.516	0.5	0.5	54.2
ExtraTrees	All	0.276	0.576	0.479	0.479*
	TF	0.592	0.810	0.730	0.708
	TFIDF	0.489	0.711	0.688	0.688
	Chi2	0.465	0.678	0.686	0.667*
	PMI	0.443	0.719	0.617	0.625*
Decision Tree	All	0.316	0.562	0.562	0.562*
	TF	0.340	0.583	0.583	0.583*
	TFIDF	0.293	0.542	0.542	0.542*
	Chi2	0.191	0.438	0.438	0.438*
	PMI	0.365	0.604	0.604	0.604*
Gaussian NB	All	0.191	0.438	0.438	0.438*
	TF	0.258	0.516	0.500	0.500*
	TFIDF	0.407	0.651	0.625	0.625*
	Chi2	0.175	0.421	0.417	0.417*
	PMI	0.192	0.462	0.417	0.417
Bernoulli NB	All	0.332	0.569	0.583	0.583*
	TF	0.641	0.832	0.771	**0.771**
	TFIDF	0.539	0.760	0.708	0.708
	Chi2	0.463	0.717	0.646	0.646*
	PMI	0.492	0.715	0.688	0.688
Multi. NB	All	0.353	0.707	0.500	0.500*
	TF	0.301	0.602	0.500	0.500*
	TFIDF	0.412	0.707	0.583	0.583*
	Chi2	0.354	0.708	0.500	0.500*
	PMI	0.339	0.679	0.500	0.500*
MLP	All	0.322	0.618	0.521	0.521*
	TF	0.323	0.646	0.500	0.500*
	TFIDF	0.464	0.675	0.688	0.688
	Chi2	0.499	0.748	0.667	0.667*
	PMI	0.319	0.637	0.500	0.500*
SGD	All	0.352	0.703	0.500	0.500*
	TF	0.311	0.622	0.500	0.500*
	TFIDF	0.592	0.790	0.750	0.750
	Chi2	0.357	0.714	0.500	0.500*
	PMI	0.346	0.693	0.500	0.500*
LDA	All	0.195	0.408	0.479	0.479 *
	TF	0.584	0.800	0.729	0.729
	TFIDF	0.206	0.495	0.417	0.417*
	Chi2	0.354	0.628	0.562	0.562*
	PMI	0.260	0.500	0.521	0.521*
Random Forest	All	0.441	0.674	0.655	0.667*
	TF	0.498	0.756	0.659	0.667*
	TFIDF	0.455	0.709	0.642	0.667*
	Chi2	0.456	0.766	0.595	0.583*
	PMI	0.557	0.754	0.738	0.750
AdaBoost	All	0.411	0.658	0.625	0.625*
	TF	0.407	0.674	0.604	0.604*
	TFIDF	0.538	0.783	0.688	0.688
	Chi2	0.494	0.719	0.688	0.688
	PMI	0.448	0.717	0.625	0.625*
Bagging	All	0.374	0.627	0.597	0.604*
	TF	0.533	0.756	0.705	0.688
	TFIDF	0.225	0.508	0.443	0.458*
	Chi2	0.274	0.527	0.520	0.521*
	PMI	0.324	0.575	0.564	0.542*
Gradient Boosting	All	0.405	0.648	0.625	0.625*
	TF	0.480	0.720	0.667	0.667*
	TFIDF	0.395	0.632	0.625	0.625*
	Chi2	0.354	0.630	0.562	0.562*
	PMI	0.469	0.703	0.667	0.667*
XGB	All	0.375	0.601	0.625	0.625*
	TF	0.326	0.625	0.521	0.521 *
	TFIDF	0.368	0.609	0.604	0.604*
	Chi2	0.310	0.595	0.521	0.521*
	PMI	0.470	0.684	0.688	0.688
Logistic Reg	All	0.442	0.707	0.625	0.625*
	TF	0.359	0.615	0.583	0.583*
	TFIDF	0.581	0.797	0.729	0.729
	Chi2	0.443	0.708	0.625	0.625*
	PMI	0.448	0.694	0.646	0.646*

Dutch reviews 2014

Model	Feature	FS	AUC	c@1	Acc
KNN	All	0.176	0.420	0.420	0.420*
	TF	0.314	0.581	0.540	0.540
	TFIDF	0.196	0.408	0.480	0.480*
	Chi2	0.252	0.504	0.500	0.500*
	PMI	0.251	0.483	0.520	0.520*
SVM	All	0.237	0.494	0.478	0.460*
	TF	0.150	0.362	0.416	0.420*
	TFIDF	0.138	0.376	0.367	0.400*
	Chi2	0.221	0.470	0.469	0.460*
	PMI	0.188	0.419	0.449	0.460*
ExtraTrees	All	0.210	0.496	0.424	0.460*
	TF	0.189	0.413	0.458	0.440*
	TFIDF	0.197	0.422	0.466	0.460*
	Chi2	0.275	0.539	0.510	0.500*
	PMI	0.147	0.361	0.408	0.400*
Decision Tree	All	0.314	0.560	0.560	0.560
	TF	0.292	0.540	0.540	0.540
	TFIDF	0.270	0.520	0.520	0.520*
	Chi2	0.270	0.520	0.520	0.520*
	PMI	0.250	0.500	0.500	0.500*
Gaussian NB	All	0.250	0.500	0.500	0.500*
	TF	0.190	0.432	0.440	0.440*
	TFIDF	0.218	0.437	0.500	0.500*
	Chi2	0.230	0.478	0.480	0.480 *
	PMI	0.242	0.485	0.500	0.500*
Bernoulli NB	All	0.210	0.419	0.500	0.500*
	TF	0.137	0.326	0.420	0.420*
	TFIDF	0.220	0.459	0.480	0.480*
	Chi2	0.252	0.525	0.480	0.480*
	PMI	0.125	0.330	0.380	0.380*
Multi. NB	All	0.229	0.477	0.480	0.480*
	TF	0.169	0.384	0.440	0.440*
	TFIDF	0.183	0.397	0.460	0.460*
	Chi2	0.214	0.466	0.460	0.460*
	PMI	0.216	0.432	0.500	0.500*
MLP	All	0.230	0.480	0.480	0.480*
	TF	0.178	0.387	0.460	0.460*
	TFIDF	0.143	0.341	0.420	0.420*
	Chi2	0.261	0.483	0.540	0.540
	PMI	0.161	0.322	0.500	0.500*
SGD	All	0.239	0.478	0.500	0.500*
	TF	0.193	0.386	0.500	0.500*
	TFIDF	0.124	0.344	0.360	0.360*
	Chi2	0.241	0.482	0.500	0.500*
	PMI	0.214	0.429	0.500	0.500*
LDA	All	0.331	0.613	0.540	0.540
	TF	0.207	0.450	0.460	0.460*
	TFIDF	0.219	0.475	0.460	0.460*
	Chi2	0.275	0.528	0.520	0.520*
	PMI	0.202	0.482	0.420	0.420*
Random Forest	All	0.235	0.494	0.475	0.460*
	TF	0.248	0.487	0.510	0.500*
	TFIDF	0.192	0.430	0.445	0.460*
	Chi2	0.319	0.569	0.562	0.560
	PMI	0.209	0.445	0.469	0.480*
AdaBoost	All	0.326	0.582	0.560	0.56*0
	TF	0.224	0.467	0.480	0.480*
	TFIDF	0.228	0.475	0.480	0.480*
	Chi2	0.333	0.616	0.540	0.540
	PMI	0.145	0.362	0.400	0.400*
Bagging	All	0.201	0.458	0.439	0.440*
	TF	0.169	0.422	0.401	0.380*
	TFIDF	0.191	0.446	0.428	0.460*
	Chi2	0.371	0.618	0.600	**0.600**
	PMI	0.233	0.513	0.454	0.500*
Gradient Boosting	All	0.269	0.538	0.500	0.500*
	TF	0.225	0.469	0.480	0.480*
	TFIDF	0.184	0.419	0.440	0.440*
	Chi2	0.285	0.549	0.520	0.520*
	PMI	0.186	0.387	0.480	0.480*
XGB	All	0.308	0.570	0.540	0.540
	TF	0.245	0.490	0.500	0.500*
	TFIDF	0.131	0.363	0.360	0.360*
	Chi2	0.294	0.589	0.500	0.500*
	PMI	0.226	0.470	0.480	0.480*
Logistic Reg	All	0.227	0.494	0.460	0.460*
	TF	0.156	0.389	0.400	0.400*
	TFIDF	0.145	0.381	0.380	0.380*
	Chi2	0.210	0.477	0.440	0.440*
	PMI	0.197	0.427	0.460	0.460*

Table 49: PAN 2014 Author verification results: part 3.

English 2015

Model	Feature	FS	AUC	c@1	Acc
KNN	All	0.351	0.629	0.558	0.558*
	TF	0.368	0.643	0.572	0.572*
	TFIDF	0.313	0.579	0.540	0.540*
	Chi2	0.391	0.662	0.590	0.590*
	PMI	0.421	0.664	0.634	0.634*
SVM	All	0.483	0.763	0.633	0.622*
	TF	0.487	0.770	0.632	0.622*
	TFIDF	0.540	0.760	0.710	0.670
	Chi2	0.509	0.775	0.657	0.620 *
	PMI	0.546	0.768	0.711	0.706
ExtraTrees	All	0.309	0.565	0.547	0.544 *
	TF	0.417	0.673	0.620	0.620*
	TFIDF	0.463	0.719	0.645	0.620*
	Chi2	0.374	0.625	0.598	0.588*
	PMI	0.329	0.585	0.563	0.566*
Decision Tree	All	0.281	0.530	0.530	0.530*
	TF	0.246	0.496	0.496	0.496*
	TFIDF	0.325	0.570	0.570	0.570*
	Chi2	0.298	0.546	0.546	0.546*
	PMI	0.362	0.602	0.602	0.602*
Gaussian NB	All	0.178	0.422	0.422	0.422*
	TF	0.209	0.456	0.458	0.458*
	TFIDF	0.494	0.727	0.680	0.680
	Chi2	0.214	0.457	0.468	0.468*
	PMI	0.250	0.508	0.492	0.492*
Bernoulli NB	All	0.173	0.397	0.436	0.436*
	TF	0.234	0.479	0.488	0.488*
	TFIDF	0.336	0.589	0.570	0.570*
	Chi2	0.187	0.409	0.458	0.458*
	PMI	0.219	0.470	0.466	0.466*
Multi. NB	All	0.375	0.750	0.500	0.500*
	TF	0.362	0.725	0.500	0.500*
	TFIDF	0.529	0.778	0.680	0.680
	Chi2	0.376	0.752	0.500	0.500*
	PMI	0.362	0.725	0.500	0.500*
MLP	All	0.427	0.684	0.624	0.624*
	TF	0.364	0.728	0.500	0.500*
	TFIDF	0.509	0.749	0.680	0.680
	Chi2	0.472	0.757	0.624	0.624*
	PMI	0.453	0.722	0.628	0.628*
SGD	All	0.372	0.745	0.500	0.500*
	TF	0.420	0.732	0.574	0.574*
	TFIDF	0.474	0.729	0.650	0.650
	Chi2	0.373	0.747	0.500	0.500*
	PMI	0.366	0.731	0.500	0.500*
LDA	All	0.228	0.476	0.480	0.480*
	TFIDF	0.294	0.542	0.542	0.542*
	TFIDF	0.416	0.682	0.610	0.610*
	Chi2	0.357	0.616	0.580	0.580*
	PMI	0.260	0.509	0.510	0.510*
Random Forest	All	0.382	0.621	0.616	0.606*
	TF	0.477	0.721	0.662	0.654
	TFIDF	0.521	0.767	0.678	0.660
	Chi2	0.496	0.732	0.677	0.672
	PMI	0.397	0.658	0.604	0.600*
AdaBoost	All	0.390	0.657	0.594	0.594*
	TF	0.401	0.643	0.624	0.624*
	TFIDF	0.417	0.662	0.630	0.630*
	Chi2	0.488	0.732	0.666	0.666
	PMI	0.395	0.648	0.610	0.610*
Bagging	All	0.275	0.539	0.510	0.524*
	TF	0.394	0.658	0.599	0.616*
	TFIDF	0.343	0.596	0.575	0.620*
	Chi2	0.419	0.671	0.625	0.616*
	PMI	0.338	0.599	0.565	0.584*
Gradient Boosting	All	0.443	0.697	0.636	0.636
	TF	0.430	0.671	0.640	0.640
	TFIDF	0.524	0.770	0.680	0.680
	Chi2	0.461	0.713	0.646	0.646
	PMI	0.435	0.693	0.628	0.628*
XGB	All	0.437	0.689	0.634	0.634*
	TF	0.480	0.730	0.658	0.658
	TFIDF	0.437	0.672	0.650	0.650
	Chi2	0.533	0.767	0.694	0.694
	PMI	0.423	0.682	0.620	0.620*
Logistic Reg	All	0.440	0.743	0.592	0.592*
	TF	0.425	0.723	0.588	0.588*
	TFIDF	0.579	0.793	0.730	**0.730**
	Chi2	0.442	0.746	0.592	0.592*
	PMI	0.478	0.719	0.664	0.664

Greek 2015

Model	Feature	FS	AUC	c@1	Acc
KNN	All	0.293	0.585	0.500	0.500*
	TF	0.332	0.582	0.570	0.570*
	TFIDF	0.328	0.565	0.580	0.580*
	Chi2	0.263	0.536	0.49	0.49*
	PMI	0.389	0.638	0.610	0.610
SVM	All	0.298	0.547	0.545	0.530*
	TF	0.288	0.571	0.505	0.530*
	TFIDF	0.526	0.777	0.677	**0.700**
	Chi2	0.286	0.544	0.525	0.520*
	PMI	0.386	0.638	0.638	0.590*
ExtraTrees	All	0.255	0.495	0.515	0.520*
	TF	0.367	0.613	0.599	0.590*
	TFIDF	0.447	0.682	0.655	0.650
	Chi2	0.328	0.573	0.572	0.560*
	PMI	0.287	0.574	0.500	0.490*
Decision Tree	All	0.292	0.540	0.540	0.540*
	TF	0.260	0.510	0.510	0.510*
	TFIDF	0.384	0.620	0.620	0.620
	Chi2	0.270	0.520	0.520	0.520*
	PMI	0.260	0.510	0.510	0.510*
Gaussian NB	All	0.270	0.520	0.520	0.520*
	TF	0.336	0.600	0.560	0.560*
	TFIDF	0.463	0.691	0.670	0.670
	Chi2	0.208	0.463	0.450	0.450*
	PMI	0.268	0.516	0.520	0.520*
Bernoulli NB	All	0.296	0.581	0.510	0.510*
	TF	0.230	0.500	0.460	0.460*
	TFIDF	0.276	0.521	0.530	0.530*
	Chi2	0.269	0.528	0.510	0.510*
	PMI	0.274	0.526	0.520	0.520*
Multi. NB	All	0.259	0.518	0.500	0.500*
	TF	0.266	0.532	0.500	0.500*
	TFIDF	0.404	0.748	0.540	0.540*
	Chi2	0.264	0.529	0.500	0.500*
	PMI	0.268	0.536	0.500	0.500*
MLP	All	0.297	0.550	0.540	0.540*
	TF	0.291	0.582	0.500	0.500*
	TFIDF	0.529	0.778	0.680	0.680
	Chi2	0.266	0.532	0.500	0.500*
	PMI	0.271	0.542	0.500	0.500*
SGD	All	0.262	0.525	0.500	0.500*
	TF	0.271	0.542	0.500	0.500*
	TFIDF	0.475	0.779	0.610	0.610
	Chi2	0.266	0.533	0.500	0.500*
	PMI	0.270	0.539	0.500	0.500*
LDA	All	0.264	0.518	0.510	0.510*
	TF	0.300	0.566	0.530	0.530*
	TFIDF	0.391	0.651	0.600	0.600*
	Chi2	0.268	0.525	0.510	0.510*
	PMI	0.319	0.550	0.580	0.580*
Random Forest	All	0.417	0.649	0.642	0.670
	TF	0.393	0.657	0.599	0.580*
	TFIDF	0.447	0.715	0.625	0.620
	Chi2	0.359	0.619	0.580	0.580*
	PMI	0.342	0.622	0.551	0.560*
AdaBoost	All	0.349	0.602	0.580	0.580*
	TF	0.258	0.516	0.500	0.500*
	TFIDF	0.348	0.590	0.590	0.590*
	Chi2	0.294	0.535	0.550	0.550*
	PMI	0.304	0.552	0.550	0.550*
Bagging	All	0.275	0.522	0.526	0.550*
	TF	0.261	0.539	0.484	0.490*
	TFIDF	0.235	0.509	0.463	0.520*
	Chi2	0.292	0.550	0.531	0.520*
Gradient Boosting	PMI	0.275	0.522	0.526	0.550*
	All	0.304	0.553	0.550	0.550*
	TF	0.286	0.540	0.530	0.530*
	TFIDF	0.400	0.667	0.600	0.600*
	Chi2	0.299	0.554	0.540	0.540*
	PMI	0.355	0.592	0.600	0.600*
XGB	All	0.258	0.527	0.490	0.490*
	TF	0.304	0.562	0.540	0.540*
	TFIDF	0.388	0.626	0.620	0.620
	Chi2	0.278	0.545	0.510	0.510*
	PMI	0.311	0.566	0.550	0.550*
Logistic Reg	All	0.264	0.518	0.510	0.510*
	TF	0.292	0.541	0.540	0.540*
	TFIDF	0.550	0.798	0.690	0.690
	Chi2	0.266	0.521	0.510	0.510*
	PMI	0.293	0.533	0.550	0.550*

Table 50: PAN 2015 Author verification results.

Dutch 2015

Model	Feature	FS	AUC	c@1	Acc
KNN	All	0.357	0.641	0.558	0.558*
	TF	0.356	0.611	0.582	0.582*
	TFIDF	0.335	0.576	0.582	0.582*
	Chi2	0.385	0.635	0.606	0.606*
	PMI	0.308	0.553	0.558	0.558*
SVM	All	0.443	0.674	0.656	0.624
	TF	0.429	0.679	0.633	0.630
	TFIDF	0.346	0.613	0.564	0.564*
	Chi2	0.383	0.639	0.599	0.606*
	PMI	0.388	0.644	0.602	0.582*
ExtraTrees	All	0.524	0.747	0.701	0.697
	TF	0.465	0.698	0.666	0.661
	TFIDF	0.415	0.663	0.625	0.606*
	Chi2	0.396	0.638	0.621	0.624
	PMI	0.483	0.711	0.679	0.679
Decision Tree	All	0.332	0.576	0.576	0.576*
	TF	0.318	0.564	0.564	0.564*
	TFIDF	0.331	.576	.576	.576*
	Chi2	0.331	0.575	0.576	0.576*
	PMI	0.353	0.594	0.594	0.594*
Gaussian NB	All	0.201	0.448	0.448	0.448*
	TF	0.402	0.645	0.624	0.624
	TFIDF	0.392	0.653	0.600	0.600*
	Chi2	0.239	0.480	0.497	0.497*
	PMI	0.298	0.553	0.539	0.539*
Bernoulli NB	All	0.417	0.656	0.636	0.636
	TF	0.414	0.638	0.648	0.648
	TFIDF	0.392	0.646	0.606	0.606*
	Chi2	0.370	0.623	0.594	0.594*
	PMI	0.439	0.703	0.624	0.624
Multi. NB	All	0.315	0.633	0.497	0.497*
	TF	0.338	0.627	0.539	0.539*
	TFIDF	0.382	0.618	0.618	0.618*
	Chi2	0.306	0.608	0.503	0.503*
	PMI	0.368	0.608	0.606	0.606*
MLP	All	0.417	0.674	0.618	0.618*
	TF	0.351	0.624	0.564	0.564*
	TFIDF	0.371	0.619	0.600	0.600*
	Chi2	0.392	0.641	0.612	0.612*
	PMI	0.285	0.567	0.503	0.503*
SGD	All	0.303	0.610	0.497	0.497*
	TF	0.316	0.628	0.503	0.503*
	TFIDF	0.380	0.627	0.606	0.606*
	Chi2	0.298	0.599	0.497	0.497*
	PMI	0.295	0.593	0.497	0.497*
LDA	All	0.272	0.522	0.521	0.521*
	TF	0.320	0.581	0.552	0.552*
	TFIDF	0.373	0.622	0.600	0.600*
	Chi2	0.362	0.609	0.594	0.594*
	PMI	0.243	0.489	0.497	0.497*
Random Forest	All	0.434	0.690	0.629	0.600*
	TF	0.513	0.722	0.710	0.691
	TFIDF	0.500	0.732	0.683	0.679
	Chi2	0.402	0.657	0.612	0.624
	PMI	0.518	0.755	0.687	0.685
AdaBoost	All	0.428	0.672	0.636	0.636
	TF	0.424	0.672	0.630	0.630
	TFIDF	0.473	0.691	0.685	0.685
	Chi2	0.312	0.560	0.558	0.558*
	PMI	0.421	0.662	0.636	0.636
Bagging	All	0.500	0.706	0.709	0.642
	TF	0.554	0.767	0.722	**0.715**
	TFIDF	0.395	0.654	0.604	0.588*
	Chi2	0.374	0.628	0.595	0.582*
	PMI	0.447	0.696	0.642	0.618*
Gradient Boosting	All	0.502	0.714	0.703	0.703
	TF	0.441	0.686	0.642	0.642
	TFIDF	0.421	0.674	0.624	0.624
	Chi2	0.372	0.633	0.588	0.588*
	PMI	0.459	0.708	0.648	0.648
XGB	All	0.427	0.671	0.636	0.636
	TF	0.446	0.675	0.661	0.661
	TFIDF	0.440	0.666	0.661	0.661
	Chi2	0.389	0.629	0.618	0.618*
	PMI	0.520	0.747	0.697	0.697
Logistic Reg	All	0.383	0.625	0.612	0.612*
	TF	0.399	0.633	0.630	0.630
	TFIDF	0.344	0.604	0.570	0.570*
	Chi2	0.359	0.605	0.594	0.594*
	PMI	0.361	0.607	0.594	0.594*

Spanish 2015

Model	Feature	FS	AUC	c@1	Acc
KNN	All	0.375	0.695	0.540	0.540*
	TF	0.317	0.598	0.530	0.530*
	TFIDF	0.414	0.739	0.560	0.560*
	Chi2	0.391	0.698	0.560	0.560*
	PMI	0.334	0.586	0.570	0.570*
SVM	All	0.594	0.787	0.755	0.700
	TF	0.475	0.710	0.670	0.640
	TFIDF	0.445	0.711	0.626	0.630
	Chi2	0.515	0.758	0.680	0.670
	PMI	0.465	0.715	0.650	0.620*
ExtraTrees	All	0.354	0.708	0.500	0.500*
	TF	0.561	0.841	0.667	0.660
	TFIDF	0.429	0.714	0.602	0.600*
	Chi2	0.466	0.801	0.581	0.570*
	PMI	0.437	0.771	0.567	0.570*
Decision Tree	All	0.325	0.570	0.570	0.570*
	TF	0.372	0.610	0.610	0.610*
	TFIDF	0.260	0.510	0.510	0.510*
	Chi2	0.325	0.570	0.570	0.570*
	PMI	0.348	0.590	0.590	0.590*
Gaussian NB	All	0.250	0.500	0.500	0.500*
	TF	0.447	0.709	0.630	0.630
	TFIDF	0.311	0.610	0.510	0.510*
	Chi2	0.250	0.500	0.500	0.500*
	PMI	0.230	0.470	0.490	0.490*
Bernoulli NB	All	0.327	0.654	0.500	0.500*
	TF	0.319	0.550	0.580	0.580*
	TFIDF	0.287	0.574	0.500	0.500*
	Chi2	0.318	0.635	0.500	0.500*
	PMI	0.275	0.585	0.470	0.470*
Multi. NB	All	0.342	0.684	0.500	0.500*
	TF	0.294	0.588	0.500	0.500*
	TFIDF	0.322	0.632	0.510	0.510*
	Chi2	0.323	0.646	0.500	0.500*
	PMI	0.336	0.672	0.500	0.500*
MLP	All	0.455	0.827	0.550	0.550*
	TF	0.311	0.623	0.500	0.500*
	TFIDF	0.436	0.703	0.620	0.620*
	Chi2	0.354	0.707	0.500	0.500*
	PMI	0.377	0.753	0.500	0.500*
LSGD	All	0.329	0.658	0.500	0.500*
	TF	0.299	0.598	0.500	0.500*
	TFIDF	0.439	0.708	0.620	0.620*
	Chi2	0.351	0.650	0.540	0.540*
	PMI	0.340	0.680	0.500	0.500 *
LDA	All	0.355	0.624	0.570	0.570 *
	TF	0.356	0.636	0.560	0.560*
	TFIDF	0.345	0.594	0.580	0.580*
	Chi2	0.459	0.676	0.680	0.680
	PMI	0.328	0.575	0.570	0.570*
Random Forest	All	0.437	0.742	0.589	0.550*
	TF	0.485	0.739	0.657	0.660
	TFIDF	0.336	0.616	0.545	0.550*
	Chi2	0.520	0.761	0.683	0.680
	PMI	0.493	0.749	0.659	0.650
AdaBoost	All	0.556	0.772	0.720	**0.720**
	TF	0.464	0.737	0.630	0.630
	TFIDF	0.320	0.581	0.550	0.550*
	Chi2	0.497	0.742	0.670	0.670
	PMI	0.495	0.762	0.650	0.650
Bagging	All	0.408	0.657	0.621	0.650
	TF	0.327	0.599	0.546	0.570*
	TFIDF	0.251	0.498	0.503	0.500*
	Chi2	0.516	0.729	0.708	0.660*
	PMI	0.353	0.623	0.567	0.590*
Gradient Boosting	All	0.345	0.594	0.580	0.580*
	TF	0.368	0.624	0.590	0.590*
	TFIDF	0.239	0.488	0.490	0.490*
	Chi2	0.322	0.575	0.560	0.560*
	PMI	0.441	0.700	0.630	0.630
XGB	All	0.402	0.660	0.610	0.610*
	TF	0.394	0.656	0.600	0.600*
	TFIDF	0.285	0.538	0.530	0.530*
	Chi2	0.415	0.649	0.640	0.640
	PMI	0.416	0.670	0.620	0.620*
Logistic Reg	All	0.384	0.663	0.580	0.580*
	TF	0.315	0.595	0.530	0.530*
	TFIDF	0.382	0.659	0.580	0.580*
	Chi2	0.371	0.651	0.570	0.570*
	PMI	0.414	0.678	0.610	0.610*

Table 51: PAN 2015 Author verification results.

English 2021

Model	Feature	AUC	c@1	F05U	F1	Brier	FS	Accuracy
KNN	All	0.644	0.569	0.589	0.679	0.670	0.630	0.569*
	TF	0.763	0.688	0.675	0.731	0.778	0.727	0.688*
	TFIDF	0.758	0.655	0.646	0.724	0.753	0.707	0.655*
	Chi2	0.720	0.651	0.645	0.703	0.753	0.694	0.65*1
	PMI	0.744	0.677	0.669	0.711	0.776	0.715	0.677*
SVM	All	0.847	0.775	0.774	0.767	0.796	0.792	0.766
	TF	0.877	0.798	0.798	0.797	0.858	0.826	0.796
	TFIDF	0.886	0.814	0.821	0.809	0.864	0.839	**0.812**
	Chi2	0.826	0.749	0.747	0.748	0.824	0.779	0.749
	PMI	0.844	0.762	0.766	0.756	0.838	0.793	0.760
ExtraTrees	All	0.818	0.732	0.741	0.686	0.784	0.752	0.716
	TF	0.875	0.796	0.797	0.792	0.827	0.817	0.792
	TFIDF	0.858	0.782	0.783	0.776	0.822	0.804	0.776
	Chi2	0.850	0.774	0.773	0.770	0.830	0.800	0.770
	PMI	0.843	0.770	0.768	0.769	0.827	0.795	0.767
Decision Tree	All	0.632	0.632	0.632	0.628	0.632	0.631	0.632*
	TF	0.670	0.670	0.670	0.670	0.670	0.670	0.670*
	TFIDF	0.634	0.634	0.634	0.634	0.634	0.634	0.634*
	Chi2	0.650	0.650	0.650	0.647	0.650	0.649	0.650*
	PMI	0.649	0.649	0.649	0.647	0.649	0.648	0.649*
Gaussian NB	All	0.417	0.417	0.410	0.405	0.417	0.413	0.417*
	TF	0.795	0.714	0.704	0.735	0.731	0.736	0.714
	TFIDF	0.838	0.751	0.741	0.764	0.764	0.772	0.751
	Chi2	0.478	0.476	0.435	0.394	0.489	0.455	0.476*
	PMI	0.498	0.498	0.553	0.659	0.498	0.541	0.498*
Bernoulli NB	All	0.666	0.616	0.616	0.597	0.638	0.626	0.616*
	TF	0.750	0.683	0.686	0.676	0.790	0.717	0.683*
	TFIDF	0.687	0.631	0.632	0.590	0.747	0.657	0.631*
	Chi2	0.642	0.608	0.609	0.623	0.720	0.641	0.608*
	PMI	0.728	0.665	0.663	0.677	0.780	0.703	0.665*
Multi. NB	All	0.390	0.407	0.421	0.431	0.409	0.412	0.407*
	TF	0.797	0.568	0.453	0.257	0.750	0.565	0.568*
	TFIDF	0.720	0.663	0.662	0.669	0.756	0.694	0.663*
	Chi2	0.440	0.441	0.377	0.333	0.458	0.410	0.441*
	PMI	0.587	0.581	0.587	0.621	0.632	0.602	0.581*
MLP	All	0.557	0.545	0.544	0.542	0.604	0.559	0.545*
	TF	0.888	0.807	0.813	0.803	0.865	0.835	0.807
	TFIDF	0.886	0.807	0.813	0.804	0.860	0.834	0.807
	Chi2	0.587	0.588	0.591	0.607	0.623	0.599	0.588*
	PMI	0.764	0.705	0.707	0.701	0.752	0.726	0.705*
SGD	All	0.639	0.616	0.616	0.620	0.617	0.622	0.616*
	TF	0.799	0.583	0.502	0.294	0.757	0.587	0.583*
	TFIDF	0.862	0.781	0.792	0.772	0.846	0.811	0.781
	Chi2	0.764	0.708	0.720	0.682	0.708	0.716	0.708
	PMI	0.702	0.685	0.669	0.744	0.685	0.697	0.685*
LDA	All	0.883	0.801	0.801	0.801	0.856	0.828	0.801
	TF	0.886	0.801	0.797	0.804	0.861	0.830	0.801
	TFIDF	0.880	0.802	0.802	0.802	0.859	0.829	0.802
	Chi2	0.741	0.686	0.680	0.707	0.784	0.720	0.686*
	PMI	0.824	0.752	0.743	0.764	0.827	0.782	0.752
Random Forest	All	0.847	0.775	0.774	0.767	0.796	0.792	0.766
	TF	0.885	0.810	0.814	0.804	0.838	0.830	0.804
	TFIDF	0.857	0.785	0.787	0.777	0.824	0.806	0.778
	Chi2	0.852	0.778	0.778	0.775	0.832	0.803	0.774
	PMI	0.843	0.768	0.767	0.765	0.830	0.795	0.765
AdaBoost	All	0.847	0.768	0.768	0.769	0.758	0.782	0.768
	TF	0.871	0.787	0.783	0.790	0.759	0.798	0.787
	TFIDF	0.840	0.771	0.770	0.772	0.757	0.782	0.771
	Chi2	0.820	0.744	0.743	0.746	0.757	0.762	0.744
	PMI	0.816	0.737	0.735	0.740	0.756	0.757	0.737
Bagging	All	0.767	0.715	0.691	0.718	0.802	0.739	0.699*
	TF	0.823	0.771	0.748	0.781	0.828	0.790	0.755
	TFIDF	0.790	0.735	0.708	0.742	0.814	0.758	0.715
	Chi2	0.788	0.727	0.703	0.734	0.811	0.753	0.711
	PMI	0.786	0.731	0.711	0.735	0.811	0.755	0.718
Gradient Boosting	All	0.878	0.797	0.799	0.796	0.853	0.825	0.797
	TF	0.889	0.802	0.802	0.803	0.863	0.832	0.802
	TFIDF	0.864	0.786	0.788	0.784	0.847	0.814	0.786
	Chi2	0.851	0.768	0.768	0.768	0.841	0.799	0.768
	PMI	0.846	0.763	0.761	0.765	0.839	0.795	0.763
XGB	All	0.883	0.801	0.801	0.801	0.856	0.828	0.801
	TF	0.892	0.805	0.804	0.806	0.864	0.834	0.805
	TFIDF	0.864	0.785	0.785	0.785	0.846	0.813	0.785
	Chi2	0.854	0.771	0.770	0.772	0.843	0.802	0.771
	PMI	0.847	0.765	0.763	0.766	0.840	0.796	0.764
Logistic Reg	All	0.636	0.604	0.605	0.610	0.633	0.618	0.604*
	TF	0.799	0.710	0.701	0.729	0.758	0.740	0.710
	TFIDF	0.862	0.776	0.775	0.776	0.847	0.807	0.776
	Chi2	0.535	0.555	0.556	0.559	0.676	0.576	0.555*
	PMI	0.771	0.708	0.711	0.703	0.801	0.739	0.708

Table 52: PAN 2021 Author verification results.

Models	Features					
	All	TF	TFIDF	Chi2	PMI	Total
KNN	1	0	0	0	2	3
SVM	2	2	3	2	2	11
ExtraTrees	5	6	5	4	2	22
Decision Tree	4	3	2	0	3	12
Gaussian NB	0	3	5	0	0	8
Bernoulli NB	1	2	3	1	2	9
Multinomial NB	0	0	2	0	0	2
MLP	1	1	6	2	0	10
SGD	0	0	5	0	0	5
LDA	2	1	1	3	1	8
Random Forest	4	**6**	**6**	**7**	**7**	30
AdaBoost	**6**	5	3	6	3	23
Bagging	3	4	0	4	2	13
Gradient Boosting	3	5	4	4	3	19
XGB	3	3	**6**	4	2	18
Logistic Reg	1	3	5	1	1	11
Total	36	44	56	38	30	

Table 53: Analysis of performance of the algorithms and feature selection

		2013				2014			
Model	Feature	FS	AUC	c@1	Acc	FS	AUC	c@1	Acc
English	FS	1				1			
	AUC	0.874	1			0.963	1		
	c@1	0.884	0.746	1		0.946	0.839	1	
	Acc	0.818	0.665	0.939	1	0.928	0.822	0.988	1
Greek	FS	1				1			
	AUC	0.919	1			0.964	1		
	c@1	0.891	0.657	1		0.936	0.830	1	
	Acc	0.811	0.584	0.891	1	0.933	0.833	0.990	1
Spanish	FS	1				1			
	AUC	0.941	1			0.927	1		
	c@1	0.794	0.587	1		0.957	0.796	1	
	Acc	0.579	0.324	0.757	1	0.955	0.796	0.997	1
Dutch	FS					1			
	AUC					0.930	1		
	c@1					0.953	0.788	1	
	Acc					0.945	0.781	0.995	1
English novels	FS					1			
	AUC					0.925	1		
	c@1					0.928	0.730	1	
	Acc					0.929	0.732	0.997	1
Dutch reviews	FS					1			
	AUC					0.963	1		
	c@1					0.906	0.769	1	
	Acc					0.903	0.780	0.980	1

Table 54: Correlation between the various measures

2015				2021			
FS	AUC	c@1	Acc	FS	AUC	c@1	Acc
1				1			
0.923	1			0.752	1		
0.923	0.715	1		0.881	0.958	1	
0.907	0.700	0.992	1	0.880	0.958	0.999	1
1							
0.969	1						
0.942	0.839	1					
0.934	0.833	0.979	1				
1							
0.889	1						
0.891	0.588	1					
0.865	0.564	0.985	1				
1							
0.956	1						
0.964	0.851	1					
0.948	0.837	0.986	1				

Table 55: Correlation between the various measures

6 Conclusion

In this thesis we have discussed our style-based text categorization works. We believed that it is a worthwhile undertaking to investigate the complete spectrum of authorship analysis approaches and to progressively develop a stable and reliable system to achieve our objective. We first developed a system to identify an author's demographic information, such as gender, age range (10s to 20s or older), bot, and language variety, using a relevant text collection (authorship profiling task). Similarly, by comparing the writing styles of the two texts, we were able to detect whether two works (chat, threatening e-mail, dubious testimony, essays, text messages, business documents, fanfiction texts) were written by the same person (authorship verification).

Based on different strategies, we participated in the CLEF-PAN evaluation campaigns from 2019 to 2022 and used prior CLEF-PAN datasets to evaluate our methods and systems over time. We chose our methods because our systems produced outcomes that were competitive with the best-performing systems, demonstrating the efficacy of our two-step feature selection methodology. We sought to create methods that, while still being straightforward and comprehensible but still generally work well.

6.1 Summary of Contributions

Through author profiling, we can identify different author classes by analyzing the sociolectal characteristics of each particular author's style. Classification of texts into classes based on the author's stylistic preferences aids in the prediction of an author's demographic, personality, education, and social networks. In this thesis, we concentrate on the most recent and popular tasks in the NLP field, including cross-genre gender identification, age range identification, language variety identification, bot identification, and profiling of fake news and hate speech spreaders.

We also worked on the widely researched linguistic and machine learning challenge of computationally identifying the author of a given text, which has applications in fields like forensics, security, criminal and civil law, or literary studies. The authorship verification problem is an example of a computational authorship analysis task in which we must determine whether or not a sample document was written by a particular author given a collection of documents authored by that author and sometimes with a set of documents written by other persons. Although more challenging, this problem is often thought to better reflect the real-life problems associated with authorship detection because it differs from the more conventional problem of choosing who among a finite number of candidate authors for whom we are given sample writings, wrote the document in question.

We examined a variety of machine learning models and discovered that they perform differently for various datasets. Different machine learning models have been proposed to determine the target category in order to determine whether or not the same model consistently proposes the best effectiveness when taking into account similar corpora under the same circumstances. These models include logistic regression, decision trees, k-nearest neighbors, support vector machines, naive Bayes, neural networks, random forests, etc. As a result, this study evaluates the performance of 16 distinct classifiers using data from nine CLEF-PAN collections for the author profiling task. For the author verification task we perform another test with the 16 classifiers on four CLEF-PAN collections.

To minimize the feature size to a few hundred terms without significantly affecting performance compared to approaches employing all the features, we suggested a two-stage feature selection process. This original feature reduction strategy has the objective of reducing the quantity of the effective vocabulary, making training and using a classifier more efficient allowing us to substitute a complex classifier (using all features) with a simpler one. Tokens are considered according to their document

frequency (df) and word frequency difference in the two-stage feature selection technique. This preferred approach was completed with a three-fold threshold (df > 3) and a one-fold threshold (tf > 1). We designed a feature set capable of differentiating each category using these two limitations. A term frequency difference is computed from the reduced number of tokens obtained by applying df > 3, but only tokens with a term frequency greater than 1 (tf > 1) are taken into account, leaving out tokens that appear just once in the text.

Our evaluation indicates that this feature reduction produces better effectiveness than approaches using all features. Our research shows that extra trees, random forest or Gradient boost typically offer the best results. Additionally, empirical data suggests that it is possible to reduce the size of the feature set to roughly 300 utilizing χ^2 and PMI scoring functions without sacrificing effectiveness.

The act of converting text into numeric form, known as text representation, is often carried out by creating a language model. Typically, these models provide probabilities, frequencies, or other numbers to individual words, word sequences, word groups, sections of documents, or entire documents. The most popular methods include distributional semantics, 1-hot encoding, n-grams, bag-of-words, and vector semantics (tf-idf, Word2vec, GloVe). With these numerical values, which each represent a word token, we created two vectors to represent each portion of the text pair for author verification. The two vectors are then subtracted to get a representation of the document's text. The TF was first applied to the full vocabulary, then just the 300 most frequent terms, and finally the top 300 TF-IDF. Then, we utilized the χ^2 and PMI scoring function taking the top 300 of these features. Our author verification system's starting point is the difference vector.

As we have shown, a straightforward strategy can produce reliable findings quickly, whereas more complicated models may not necessarily produce outcomes that are considerably better but always take longer to complete. Furthermore, because our straightforward methods rely on word frequencies, word frequency difference, document frequencies, and document vector differences rather than a black box methodology, the outcomes can be explainable.

6.2 Future Directions

The accomplishments that are highlighted in this thesis could be extended in different directions. While we believe that the area has advanced to a high level of maturity in terms of authorship analysis performance from the perspective of laboratory experiments, it is clear that considerable work has to be done in order to implement author authorship analysis technology in the "real" world setting due to some areas that are yet to be looked into such as:

- Future study on gender identification should examine these performances in languages with less similarity to English, such as Bengali or Chinese. Additionally, studies could expand the gender gap by investigating additional datasets with photographs, images, or video as well as other Internet communication technologies (like SMS) or on particular text genres rather than just text and tweets (e.g., political speeches).
- The Internet developed into a significant application of language identification due to the huge volume of multilingual content it contains. A document's language can be ascertained using computational approaches before going through additional processing. Modern techniques for language identification produce satisfactory results above 95% accuracy for the majority of European languages (Portuguese and Spanish), as shown in Table 31. This degree of accomplishment is typical when working with languages that are members of the family of low-resource languages. Distinction based on word n-gram models typically performs well for well studied language combinations. The comparison of languages with different spellings for the same terms may also aid in language identification (British English and American English).

When compared to language pairs with identical character sets (Arabic), these languages are simpler to distinguish. Can we determine a writer's original language if English isn't their first language with this success? Can we tell a German author from a Chinese author by their English-language writing?

- Since the use of bots is growing, we should also look at profiling the various bot kinds that are in use and their primary function. An estimate states that two-thirds of tweeted links to well-known websites are provided by automated accounts rather than by actual people. By pushing a product and/or generating positive reviews, bots might raise its popularity artificially. They could also damage the reputation of rival products by giving them low ratings. When political or ideological ends are pursued, the hazard is significantly greater. Therefore, from the perspectives of marketing, forensics, and security, it is crucial to approach the detection of different types of bots from an author profiling standpoint.
- Authorship verification technologies have advanced to the point that they may now be carefully used in real-world situations to settle authorship disputes. Despite their wide spread use, none of the currently available methods has been proven to be faultless. Research should take advantage of the massive corpora accumulated through years of stylometry study in order to give a solid foundation for empirical evaluations of verification approaches; promising tactics like decision fusion should be further examined. These techniques should also be put through rigorous testing in applications that focus on security and privacy as well as hostile circumstances.

References

[1] Abbas, M., Ali, K., Memon, S., Jamali, A., Memon, S., Ahmed, A.: Multinomial naive bayes classification model for sentiment analysis (03 2019)

[2] Abiodun, O., Jantan, A., Omolara, O., Dada, K., Mohamed, N., Arshad, H.: State-of-the-art in artificial neural network applications: A survey. Heliyon 4, e00938 (11 2018)

[3] Akhtar, M., Kumar, A., Ghosal, D., Ekbal, A., Bhattacharyya, P.: A multilayer perceptron based ensemble technique for fine-grained financial sentiment analysis. pp. 540–546 (01 2017)

[4] Al-Anzi, F., AbuZeina, D.: Arabic text classification using linear discriminant analysis. pp. 1–6 (05 2017)

[5] Al-Batah, M., Mrayyen, S., Alzaqebah, M.: Arabic sentiment classification using mlp network hybrid with naive bayes algorithm. Journal of Computer Science 14, 1104–1114 (05 2019)

[6] Al-Shammari, R., Yousif, S.A.: Fake news classification using random forest and decision tree (j48) 23, 8 (12 2020)

[7] Argamon, S., Fine, J., Shimoni, A.: Gender, genre, and writing style in formal written texts. Text 23 (12 2003)

[8] Argamon, S., Koppel, M., Pennebaker, J., Schler, J.: Automatically profiling the author of an anonymous text. Commun. ACM 52, 119–123 (02 2009)

[9] Athanasiou, V., Maragoudakis, M.: A novel, gradient boosting framework for sentiment analysis in languages where nlp resources are not plentiful: A case study for modern greek. Algorithms 10, 34 (2017)

[10] Bahad, P., Saxena, P., Kamal, R.: Fake news detection using bi-directional lstm-recurrent neural network. Procedia Computer Science 165, 74–82 (02 2020)

[11] Basu, A., Watters, C., Author, M.: Support vector machines for text categorization. p. 103 (01 2003)

[12] Bentolila, I., Zhou, Y., Ismail, I., Humpleman, R.: System and method for behavioral model clustering in television usage, targeted advertising via model clustering, and preference programming based on behavioral model clusters (07 2013)

[13] Bessi, A., Ferrara, E.: Social bots distort the 2016 u.s. presidential election online discussion. First Monday 21 (11 2016)

[14] Bevendorff, J., Chulvi, B., Sarracén, G.L.D.L.P., Kestemont, M., Manjavacas, E., Markov, I., Mayerl, M., Potthast, M., Francisco Rangel, P.R., Stamatatos, E., Stein, B., Wiegmann, M., Wolska, M., Zangerle, E.: Overview of PAN 2021: Authorship Verification, Profiling Hate Speech Spreaders on Twitter and Style Change Detection. In: K. Selcuk Candan, Bogdan Ionescu, L.G.B.L.H.M.A.J.M.M.F.P.G.F.N.F. (ed.) 12th International Conference of the CLEF Association (CLEF 2021). Springer (2021)

[15] Bevendorff, J., Ghanem, B., Giachanou, A., Kestemont, M., Manjavacas, E., Markov, I., Mayerl, M., Potthast, M., Rangel, F., Rosso, P., Specht, G., Stamatatos, E., Stein, B., Wiegmann, M., Zangerle, E.: Overview of pan 2020: Authorship verification, celebrity profiling, profiling fake news spreaders on twitter, and style change detection. In: Avi Arampatzis, Evangelos Kanoulas, T.T.S.V.H.J.C.L.C.E.A.N.L.C.N.F. (ed.) 11th International Conference of the CLEF Association (CLEF 2020). Springer (September 2020), http://ceur-ws.org/Vol-2696/

[16] Bevendorff, J., Stein, B., Hagen, M., Potthast, M.: Generalizing unmasking for short texts. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers). pp. 654–659. Association for Computational Linguistics, Minneapolis, Minnesota (Jun 2019), https://aclanthology.org/N19-1068

[17] Bishop, C.: The multi-layer perceptron. Neural Networks for Pattern Recognition pp. 116–163 (01 1995)

[18] Bolonyai, F., Buda, J., Katona, E.: Bot or not: A two-level approach in author profiling notebook for pan at clef 2019 (01 2019)

[19] Bonadiman, D., Castellucci, G., Favalli, A., Romagnoli, R., Moschitti, A.: Neural sentiment analysis for a real-world application (12 2017)

[20] Bottou, L., Curtis, F.E., Nocedal, J.: Optimization methods for large-scale machine learning. SIAM Review 60 (06 2016)

[21] Bouaziz, A., Dartigues-Pallez, C., da Costa Pereira, C., Precioso, F., Lloret, P.: Short text classification using semantic random forest. pp. 288–299 (09 2014)

[22] Boukhaled, M., Ganascia, J.G.: Stylistic Features Based on Sequential Rule Mining for Authorship Attribution, pp. 159–175 (12 2017)

[23] Breiman, L.: Random forests–random features (09 2021)

[24] Brier, G.W.: Verification of forecasts expressed in terms of probability. Monthly Weather Review 78, 1–3 (1950)

[25] Bühlmann, P.: Bagging, boosting and ensemble methods. Handbook of Computational Statistics (01 2012)

[26] Celli, F., Pianesi, F., Stillwell, D., Kosinski, M.: Workshop on computational personality recognition: Shared task (07 2013)

[27] Chang, M.W., Yih, W.t., Meek, C.: Partitioned logistic regression for spam filtering. pp. 97–105 (08 2008)

[28] Chaski, C.: Forensic linguistics: An introduction to language, crime and the law 11, 298–303 (01 2004)

[29] Chen, T., Guestrin, C.: Xgboost: A scalable tree boosting system. CoRR abs/1603.02754 (2016), http://arxiv.org/abs/1603.02754

[30] Church, K., Hanks, P.: Word association norms, mutual information, and lexicography. Computational Linguistics 16 (07 2002)

[31] Diab, S.: Optimizing stochastic gradient descent in text classification based on fine-tuning hyper-parameters approach. a case study on automatic classification of global terrorist attacks. International Journal of Computer Science and Information Security, 16, 155–160 (02 2019)

[32] Dickerson, J., Kagan, V., Subrahmanian, V.: Using sentiment to detect bots on twitter: Are humans more opinionated than bots? pp. 620–627 (08 2014)

[33] Diederich, J., Kindermann, J., Leopold, E., Paass, G.: Authorship attribution with support vector machines. Applied Intelligence 19 (07 2003)

[34] Elfardy, H., Diab, M.: Sentence level dialect identification in arabic. vol. 2, pp. 456–461 (08 2013)

[35] Eswaran, P., Santhosh, M., Krishnan, A., Kumar, T.: Sentiment analysis of us airline twitter data using new adaboost approach. International Journal of Engineering and Technical Research 7, 1–3 (04 2019)

[36] Farías, D.I.H., Patti, V., Rosso, P.: Irony detection in twitter: The role of affective content. ACM Trans. Internet Technol. 16(3) (jul 2016), https://doi.org/10.1145/2930663

[37] Fauzi, M.: Random forest approach for sentiment analysis in indonesian language. Indonesian Journal of Electrical Engineering and Computer Science 12, 46–50 (10 2018)

[38] Forman, G.: An extensive empirical study of feature selection metrics for text classification [j]. Journal of Machine Learning Research - JMLR 3 (03 2003)

[39] Fortuna, P., Nunes, S.: A survey on automatic detection of hate speech in text. ACM Comput. Surv. 51(4) (jul 2018), https://doi.org/10.1145/3232676

[40] Freund, Y., Schapire, R.E.: A desicion-theoretic generalization of on-line learning and an application to boosting. In: Vitányi, P. (ed.) Computational Learning Theory. pp. 23–37. Springer Berlin Heidelberg, Berlin, Heidelberg (1995)

[41] Freund, Y., Schapire, R.E.: A desicion-theoretic generalization of on-line learning and an application to boosting. In: Vitányi, P. (ed.) Computational Learning Theory. pp. 23–37. Springer Berlin Heidelberg, Berlin, Heidelberg (1995)

[42] Friedman, J.H.: Greedy function approximation: A gradient boosting machine. The Annals of Statistics 29(5), 1189 – 1232 (2001), https://doi.org/10.1214/aos/1013203451

[43] Geurts, P., Ernst, D., Wehenkel, L.: Extremely randomized trees. Machine Learning 63, 3–42 (04 2006)

[44] Gezici, B., Bölücü, N., Kolukısa Tarhan, A., Can, B.: Neural sentiment analysis of user reviews to predict user ratings. pp. 629–634 (09 2019)

[45] Ghanem, B., Rosso, P., Rangel Pardo, F.: An emotional analysis of false information in social media and news articles. ACM Transactions on Internet Technology 20, 1–18 (04 2020)

[46] Giachanou, A., Rosso, P., Crestani, F.: Leveraging emotional signals for credibility detection. pp. 877–880 (07 2019)

[47] Gilani, Z., Farahbakhsh, R., Tyson, G., Wang, L., Crowcroft, J.: Of bots and humans (on twitter). In: Proceedings of the 2017 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2017. p. 349–354. ASONAM '17, Association for Computing Machinery, New York, NY, USA (2017), https://doi.org/10.1145/3110025.3110090

[48] Goswami, S., Sarkar, S., Rustagi, M.: Stylometric analysis of bloggers' age and gender. In: ICWSM (2009)

[49] Greff, K., Srivastava, R., Koutník, J., Steunebrink, B., Schmidhuber, J.: Lstm: A search space odyssey. IEEE transactions on neural networks and learning systems 28 (03 2015)

[50] Grimm, L.G.: Statistical Applications for the Behavioral Sciences. John Wiley Sons (1993)

[51] Guo, C., Cao, J., Zhang, X., Shu, K., Liu, H.: Dean: Learning dual emotion for fake news detection on social media. arXiv: Computation and Language (2019)

[52] Halteren, H.: Author verification by linguistic profiling: An exploration of the parameter space. TSLP 4 (01 2007)

[53] Halvani, O., Winter, C., Graner, L.: Authorship verification based on compression-models (06 2017)

[54] Hastie, T., Tibshirani, R., Friedman, J.: The elements of statistical learning: data mining, inference and prediction. Springer, 2 edn. (2009), http://www-stat.stanford.edu/ tibs/ElemStatLearn/

[55] Hirst, G., Feng, V.: Changes in style in authors with alzheimer's disease. English Studies 93, 357–370 (05 2012)

[56] Holland, S.: Principal components analysis (pca) (2008)

[57] Holmes, D.: The evolution of stylometry in humanities scholarship. Literary and Linguistic Computing 13, 111–117 (1998)

[58] Ifrim, G., Bakir, G., Weikum, G.: Fast logistic regression for text categorization with variable-length n-grams. Bing Liu, Bing; Sarawagi, Sunita; Li, Ying: KDD 2008 : proceedings of the 14th ACM KDD International Conference on Knowledge Discovery Data Mining, ACM, 354-362 (2008) (08 2008)

[59] Ikae, C., Nath, S., Savoy, J.: Unine at pan-clef 2019: Bots and gender task. In: CLEF (2019)

[60] Ikae, C., Savoy, J.: Gender identification on twitter. Journal of the Association for Information Science and Technology 73 (06 2021)

[61] Ikae, C., Savoy, J.: Gender identification on twitter. Journal of the Association for Information Science and Technology 73 (06 2021)

[62] Iqbal, F., Khan, L.A., Fung, B., Debbabi, M.: E-mail authorship verification for forensic investigation. pp. 1591–1598 (01 2010)

[63] Iqbal, F., Khan, L.A., Fung, B., Debbabi, M.: E-mail authorship verification for forensic investigation. pp. 1591–1598 (01 2010)

[64] Jadhav, S., Thepade, S.: Fake news identification and classification using dssm and improved recurrent neural network classifier. Applied Artificial Intelligence 33, 1–11 (09 2019)

[65] Jamak, A., Savatić, A., Can, M.: Principal component analysis for authorship attribution. Business Systems Research 3 (09 2012)

[66] James, G., Witten, D., Hastie, T., Tibshirani, R.: An Introduction to Statistical Learning: With Applications in R. Springer Publishing Company, Incorporated (2014)

[67] Joachims, T.: Text categorization with support vector machines. Proc. European Conf. Machine Learning (ECML'98) (01 1998)

[68] Jotheeswaran, J., Seerangan, K.: Decision tree based feature selection and multilayer perceptron for sentiment analysis. ARPN Journal of Engineering and Applied Sciences 10, 5883–5894 (01 2015)

[69] Kadam, S., Gala, A., Gehlot, P., Kurup, A., Ghag, K.: Word embedding based multinomial naive bayes algorithm for spam filtering. pp. 1–5 (08 2018)

[70] Kaliyar, R., Goswami, A., Narang, P., Sinha, S.: Fndnet- a deep convolutional neural network for fake news detection. Cognitive Systems Research 61 (06 2020)

[71] Kamel, H., Al-Tuwaijari, J.: Cancer classification using gaussian naive bayes algorithm. pp. 165–170 (06 2019)

[72] Kestemont, M., Manjavacas, E., Markov, I., Bevendorff, J., Wiegmann, M., Stamatatos, E., Potthast, M., Stein, B.: Overview of the Cross-Domain Authorship Verification Task at PAN 2020. In: Cappellato, L., Eickhoff, C., Ferro, N., Névéol, A. (eds.) CLEF 2020 Labs and Workshops, Notebook Papers. CEUR-WS.org (Sep 2020), http://ceur-ws.org/Vol-2696/

[73] Kestemont, M., Stamatatos, E., Manjavacas, E., Bevendorff, J., Potthast, M., Stein, B.: Overview of the Authorship Verification Task at PAN 2021. In: CLEF 2021 Labs and Workshops, Notebook Papers. CEUR-WS.org (2021)

[74] Kestemont, M., Stamatatos, E., Manjavacas, E., Daelemans, W., Potthast, M., Stein, B.: Overview of the cross-domain authorship attribution task at pan 2019. In: CLEF (2019)

[75] Kestemont, M., Stover, J., Koppel, M., Karsdorp, F., Daelemans, W.: Authenticating the writings of julius caesar. Expert Systems with Applications 63 (06 2016)

[76] Khamar, K.: Short text classification using knn based on distance function (2013)

[77] Khomsah, S.: Sentiment analysis on youtube comments using word2vec and random forest. Telematika 18, 61 (03 2021)

[78] Kocher, M., Savoy, J.: Evaluation of text representation schemes and distance measures for authorship linking. Digital Scholarship in the Humanities 34 (02 2018)

[79] Koppel, M.: Automatically categorizing written texts by author gender. Literary and Linguistic Computing 17, 401–412 (11 2002)

[80] Koppel, M., Schler, J., Bonchek Dokow, E.: Measuring differentiability: Unmasking pseudonymous authors. Journal of Machine Learning Research 8, 1261–1276 (06 2007)

[81] Koppel, M., Winter, Y.: Determining if two documents are written by the same author. Journal of the Association for Information Science and Technology 65 (01 2014)

[82] Kurniasari, L., Setyanto, A.: Sentiment analysis using recurrent neural network. Journal of Physics: Conference Series 1471, 012018 (02 2020)

[83] Lengkong, O., Maringka, R.: Apps rating classification on play store using gradient boost algorithm. pp. 1–5 (10 2020)

[84] Li, F., Fan, J., Wang, L., Zhang, H., Duan, R.: A method based on manifold learning and bagging for text classification (08 2011)

[85] Li, M., Fu, X., Li, D.: Diabetes prediction based on xgboost algorithm. IOP Conference Series: Materials Science and Engineering 768, 072093 (03 2020)

[86] Lukosevicius, M.: A practical guide to applying echo state networks. In: Neural Networks: Tricks of the Trade (2012)

[87] Madigan, D., Genkin, A., Lewis, D., Fradkin, D.: Bayesian multinomial logistic regression for author identification. 25th International Workshop on Bayesian Inference and Maximum Entropy Methods in Science and Engineering (11 2005)

[88] Maier, W., Gómez-Rodríguez, C.: Language variety identification in spanish tweets. In: EMNLP 2014 (2014)

[89] Malmasi, S., Evanini, K., Cahill, A., Tetreault, J., Pugh, R., Hamill, C., Napolitano, D., Qian, Y.: A Report on the 2017 Native Language Identification Shared Task. In: Proceedings of the 12th Workshop on Building Educational Applications Using NLP. Association for Computational Linguistics, Copenhagen, Denmark (September 2017)

[90] Maruf, S., Javed, K., Babri, H.: Improving text classification performance with random forests-based feature selection. Arabian Journal for Science and Engineering 41 (11 2015)

[91] Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. Proceedings of Workshop at ICLR 2013 (01 2013)

[92] Mugdha, S., Binte Mohammed, M., Salsabil, L., Anika, A., Marma, P., Hossain, Z., Shatabda, S.: A Gaussian Naive Bayesian Classifier for Fake News Detection in Bengali, pp. 283–291 (05 2021)

[93] Mullen, T., Collier, N.: Sentiment analysis using support vector machines with diverse information sources. pp. 412–418 (01 2004)

[94] Mustapha, I.B., Hasan, S., Olatunji, S.O., Shamsuddin, S.M., Kazeem, A.: Effective email spam detection system using extreme gradient boosting (2020)

[95] Neal, T., Sundararajan, K., Woodard, D.: Exploiting linguistic style as a cognitive biometric for continuous verification. pp. 270–276 (02 2018)

[96] Nguyen, D., Smith, N.A., Rosé, C.P.: Author age prediction from text using linear regression. In: Proceedings of the 5th ACL-HLT Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities. p. 115–123. LaTeCH '11, Association for Computational Linguistics, USA (2011)

[97] Nikhath, A.K., Subrahmanyam, K., Vasavi, R.: Building a k-nearest neighbor classifier for text categorization (2016)

[98] Noormanshah, W., Nohuddin, P., Zainol, Z.: Document categorization using decision tree: Preliminary study. International Journal of Engineering and Technology 7, 437–440 (12 2018)

[99] Ogdol, J.M., Samar, B.L.: Binary logistic regression based classifier for fake news (06 2018)

[100] Pardo, F.M.R., Celli, F., Rosso, P., Potthast, M., Stein, B., Daelemans, W.: Overview of the 3rd author profiling task at pan 2015. In: CLEF (2015)

[101] Pardo, F.M.R., Rosso, P.: Overview of the 7th author profiling task at pan 2019: Bots and gender profiling in twitter. In: CLEF (2019)

[102] Pardo, F.M.R., Rosso, P., Potthast, M., Stein, B.: Overview of the 5th author profiling task at pan 2017: Gender and language variety identification in twitter. In: CLEF (2017)

[103] Pardo, F.M.R., Rosso, P., Verhoeven, B., Daelemans, W., Potthast, M., Stein, B.: Overview of the 4th author profiling task at pan 2016: Cross-genre evaluations. In: CLEF (2016)

[104] Pardo, F.M.R., Giachanou, A., Ghanem, B., Rosso, P.: Overview of the 8th author profiling task at pan 2020: Profiling fake news spreaders on twitter. In: Cappellato, L., Eickhoff, C., 0001, N.F., Névéol, A. (eds.) Working Notes of CLEF 2020 - Conference and Labs of the Evaluation Forum, Thessaloniki, Greece, September 22-25, 2020. CEUR Workshop Proceedings, vol. 2696. CEUR-WS.org (2020), http://ceur-ws.org/Vol-2696/paper$_2$67.*pdf*

[105] Pardo, F.M.R., Franco-Salvador, M., Rosso, P.: A low dimensionality representation for language variety identification. ArXiv abs/1705.10754 (2016)

[106] Pardo, F.M.R., Rosso, P., y Gómez, M.M., Potthast, M., Stein, B.: Overview of the 6th author profiling task at pan 2018: Multimodal gender identification in twitter. In: CLEF (2018)

[107] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., VanderPlas, J., Passos, A., Cournapeau, D., Brucher,

M., Perrot, M., Duchesnay, E.: Scikit-learn: Machine learning in python. CoRR abs/1201.0490 (2012), http://arxiv.org/abs/1201.0490

[108] Peñas, A., Rodrigo, : A simple measure to assess non-response. vol. 1, pp. 1415–1424 (01 2011)

[109] Potha, N., Stamatatos, E.: A profile-based method for authorship verification. pp. 313–326 (05 2014)

[110] Pranckevicius, T., Marcinkevičius, V.: Application of logistic regression with part-of-the-speech tagging for multi-class text classification. pp. 1–5 (11 2016)

[111] Purwandari, K., Rahutomo, R., Sigalingging, J., Kusuma, M., Prasetyo, A., Pardamean, B.: Twitter-based text classification using svm for weather information system (08 2021)

[112] Qaiser, S., Ali, R.: Text mining: Use of tf-idf to examine the relevance of words to documents. International Journal of Computer Applications 181, 25–29 (2018)

[113] Raihan, M.: Emotion detection of twitter post using multinomial naive bayes (10 2020)

[114] Ramamurthy, R., Stenzel, R., Sifa, R., Ladi, A., Bauckhage, C.: Echo State Networks for Named Entity Recognition, pp. 110–120 (09 2019)

[115] Rangel, F., Celli, F., Rosso, P., Potthast, M., Stein, B., Daelemans, W.: Overview of the 3rd Author Profiling Task at PAN 2015. In: Cappellato, L., Ferro, N., Jones, G., San Juan, E. (eds.) Working Notes Papers of the CLEF 2015 Evaluation Labs. Lecture Notes in Computer Science, vol. 1391 (Sep 2015), http://ceur-ws.org/Vol-1391/

[116] Rangel, F., Giachanou, A., Ghanem, B., Rosso, P.: Overview of the 8th Author Profiling Task at PAN 2020: Profiling Fake News Spreaders on Twitter. In: Cappellato, L., Eickhoff, C., Ferro, N., Névéol, A. (eds.) CLEF 2020 Labs and Workshops, Notebook Papers. CEUR-WS.org (Sep 2020), http://ceur-ws.org/Vol-2696/

[117] Rangel, F., Rosso, P.: Overview of the 7th Author Profiling Task at PAN 2019: Bots and Gender Profiling. In: Cappellato, L., Ferro, N., Losada, D., Müller, H. (eds.) CLEF 2019 Labs and Workshops, Notebook Papers. CEUR-WS.org (Sep 2019), http://ceur-ws.org/Vol-2380/

[118] Rangel, F., Rosso, P., Chugur, I., Potthast, M., Trenkmann, M., Stein, B., Verhoeven, B., Daelemans, W.: Overview of the 2 nd author profiling task at pan 2014 (2014)

[119] Rangel, F., Rosso, P., Chugur, I., Potthast, M., Trenkmann, M., Stein, B., Verhoeven, B., Daelemans, W.: Overview of the 2nd Author Profiling Task at PAN 2014. In: Cappellato, L., Ferro, N., Halvey, M., Kraaij, W. (eds.) Working Notes Papers of the CLEF 2014 Evaluation Labs. Lecture Notes in Computer Science, vol. 1180 (Sep 2014), http://ceur-ws.org/Vol-1180/

[120] Rangel, F., Rosso, P., Potthast, M., Stein, B.: Overview of the 5th Author Profiling Task at PAN 2017: Gender and Language Variety Identification in Twitter. In: Cappellato, L., Ferro, N., Goeuriot, L., Mandl, T. (eds.) Working Notes Papers of the CLEF 2017 Evaluation Labs. CEUR Workshop Proceedings, vol. 1866 (Sep 2017), http://ceur-ws.org/Vol-1866/

[121] Rangel, F., Rosso, P., Verhoeven, B., Daelemans, W., Potthast, M., Stein, B.: Overview of the 4th Author Profiling Task at PAN 2016: Cross-Genre Evaluations. In: Balog, K., Cappellato, L., Ferro, N., Macdonald, C. (eds.) Working Notes Papers of the CLEF 2016 Evaluation Labs. Lecture Notes in Computer Science, vol. 1609 (Sep 2016), http://ceur-ws.org/Vol-1609/

[122] Rangel Pardo, F., Rosso, P., Chugur, I., Potthast, M., Trenkmann, M., Stein, B., Verhoeven, B., Daelemans, W.: Overview of the 2nd author profiling task at pan 2014. CEUR Workshop Proceedings 1180, 898–927 (01 2013)

[123] Rani, V., Rani, K.: Identification of Ontologies of Prediabetes Using SVM Sentiment Analysis, pp. 535–550 (01 2020)

[124] Rashkin, H., Choi, E., Jang, J., Volkova, S., Choi, Y.: Truth of varying shades: Analyzing language in fake news and political fact-checking. pp. 2931–2937 (01 2017)

[125] Reddy, T., Bulusu, V.v., Reddy, V.: A survey on authorship profiling techniques 11, 3092–3102 (03 2016)

[126] Rokach, L., Maimon, O.: Decision Trees, vol. 6, pp. 165–192 (01 2005)

[127] Sadat, F., Kazemi, F., Farzindar, A.: Automatic identification of arabic language varieties and dialects in social media. pp. 22–27 (01 2014)

[128] Saigal, P., Khanna, V.: Multi-category news classification using support vector machine based classifiers. SN Applied Sciences 2 (03 2020)

[129] Salton, G., Buckley, C.: Term-weighting approaches in automatic text retrieval. Inf. Process. Manag. 24, 513–523 (1988)

[130] Santhanam, R., Uzir, N., Raman, S., Banerjee, S.: Experimenting xgboost algorithm for prediction and classification of different datasets (03 2017)

[131] Savoy, J.: Comparative evaluation of term selection functions for authorship attribution. Digital Scholarship in the Humanities 30(2), 246–261 (08 2013), https://doi.org/10.1093/llc/fqt047

[132] Savoy, J.: Machine learning methods for stylometry: Authorship attribution and author profiling. Springer International Publishing (2020)

[133] Saxena, A., Das, G., Sain, A.: Mining criminal dataset using gradient boosting algorithm. SSRN Electronic Journal (01 2019)

[134] Schaetti, N.: Behaviors of reservoir computing models for textual documents classification (07 2019)

[135] Schler, J., Koppel, M., Argamon, S., Pennebaker, J.: Effects of age and gender on blogging. In: AAAI Spring Symposium: Computational Approaches to Analyzing Weblogs (2006)

[136] Schler, J., Koppel, M., Argamon, S., Pennebaker, J.: Effects of age and gender on blogging. pp. 199–205 (01 2006)

[137] Schler, J., Koppel, M., Argamon, S., Pennebaker, J.: Effects of age and gender on blogging. pp. 199–205 (01 2006)

[138] Sebastiani, F.: Machine learning in automated text categorization. ACM Computing Surveys 34, 1–47 (04 2001)

[139] Shah, F., Ahmed, S.: Fake review detection using principal component analysis and active learning. International Journal of Computer Applications 178, 42–48 (09 2019)

[140] Shanahan, J., Roma, N.: Improving svm text classification performance through threshold adjustment. vol. 2837, pp. 361–372 (09 2003)

[141] Sharaff, A., Gupta, H.: Extra-Tree Classifier with Metaheuristics Approach for Email Classification, pp. 189–197 (05 2019)

[142] Sharif, O., Hoque, M., Kayes, A.S.M., Nowrozy, R., Sarker, I.: Detecting suspicious texts using machine learning techniques (07 2020)

[143] Sherstinsky, A.: Fundamentals of recurrent neural network (rnn) and long short-term memory (lstm) network. Physica D: Nonlinear Phenomena 404, 132306 (03 2020)

[144] Shu, K., Wang, S., Liu, H.: Understanding user profiles on social media for fake news detection. In: 2018 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR). pp. 430–435 (2018)

[145] Singh, M., Bhatt, M., Bedi, H., Mishra, U.: Performance of bernoulli's naive bayes classifier in the detection of fake news. Materials Today: Proceedings (12 2020)

[146] Stella, M., Ferrara, E., Domenico, M.D.: Bots increase exposure to negative and inflammatory content in online social systems. Proceedings of the National Academy of Sciences of the United States of America 115, 12435 – 12440 (2018)

[147] Stover, J., Winter, Y., Koppel, M., Kestemont, M.: Computational authorship verification method attributes a new work to a major 2nd century african author. Journal of the Association for Information Science and Technology 67 (05 2015)

[148] Suman, C., Saha, S., Bhattacharyya, P., Chaudhari, R.: Emoji helps! a multi-modal siamese architecture for tweet user verification. Cognitive Computation 13 (03 2021)

[149] Syamala, M., Nalini, N.: A filter based improved decision tree sentiment classification model for realtime amazon product review data. International Journal of Intelligent Engineering and Systems 13, 191–202 (02 2020)

[150] Taheri, R., Javidan, R.: Spam filtering in sms using recurrent neural networks (11 2018)

[151] Torkkola, K.: Linear discriminant analysis in document classification. IEEE TextDM 2001 (12 2001)

[152] Wang, W.Y.: "liar, liar pants on fire": A new benchmark dataset for fake news detection. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). pp. 422–426. Association for Computational Linguistics, Vancouver, Canada (Jul 2017), https://aclanthology.org/P17-2067

[153] Wen, S., Wei, H., Yang, Y., Guo, Z., Zeng, Z., Huang, T., Chen, Y.: Memristive lstm network for sentiment analysis. IEEE Transactions on Systems, Man, and Cybernetics: Systems PP, 1–11 (04 2019)

[154] Witten, I.H., Frank, E., Hall, M.A.: Data Mining: Practical Machine Learning Tools and Techniques. Morgan Kaufmann Series in Data Management Systems, Morgan Kaufmann, Amsterdam, 3 edn. (2011), http://www.sciencedirect.com/science/book/9780123748560

[155] Xanthopoulos, P., Pardalos, P., Trafalis, T.: Linear Discriminant Analysis, pp. 27–33 (01 2013)

[156] Xipeng, Z., Xiong, G., Yuexiang, H., Zhu, F., Dong, X., Nyberg, T.: A method of sms spam filtering based on adaboost algorithm (07 2016)

[157] Yang, W., Yuan, T., Wang, L.: Micro-blog sentiment classification method based on the personality and bagging algorithm. Future Internet 12, 75 (04 2020)

[158] Yang, Y., Liu, X.: A re-examination of text categorization methods. Proceedings of the 22nd SIGIR, New York, NY, USA (01 2003)