Author Profiling: Gender Prediction from Tweets and Images

Notebook for PAN at CLEF 2018

Yaakov HaCohen-Kerner[1], Yair Yigal[1], Elyashiv Shayovitz[1], Daniel Miller[1], and Toby Breckon[2]

[1]Department of Computer Science, Jerusalem College of Technology, Lev Academic Center
21 Havaad Haleumi St., P.O.B. 16031, 9116001 Jerusalem, Israel
[2]Department of Computer Science, the Palatine Centre, Durham University, Stockton Road, Durha, DH1 3LE, UK

kerner@jct.ac.il, yigalyairn@gmail.com, elyashiv12@gmail.com, 3danmi@gmail.com, toby.breckon@durham.ac.uk

Abstract. Author profiling deals with identification of various details about the author of the text (e.g., age, cultural background, gender, native language, personality). In this paper, we describe the participation of our teams (yigal18 and miller18, both teams contain the same people, but in another order) in the PAN 2018 shared task on author profiling, identifying authors' gender where for each author, 100 tweets and 10 images are provided. The authors were grouped by the language of their tweets: English, Spanish, and Arabic. In this paper, we describe our pre-processing, feature sets, machine learning methods and accuracy results. The best results using the textual features were achieved using the MLP method after applying the L normalization and using 9,000 word unigrams for English, 10,000 word unigrams and one orthographic feature for Spanish, and 7,000 word unigrams and one orthographic feature for Arabic. We also tried various additional feature sets, including style-based feature sets. In most of the cases, these features did not improve the results and in a few cases even hurt the results. The best result (61.54%) for the visual features was obtained by the LR method using all the features (SIFT & Color & VGG) and the best basic feature set is the VGG. The best result for the combined features was achieved using model2 (miller18) with 0.75 as a weight to the best textual model and a weight of 0.25 for NN Classifier (Keras) using only the 1000 VGG features.

Keywords: Author Profiling, Gender Classification, Content-based Features, Style-based Features, Images, Supervised Machine Learning, Tweets, Visual Features.

1 Introduction

Author profiling deals with analysis of a given text while inferring various information about the author of the text (e.g., age, cultural background, gender, native language, personality). This problem is of growing importance all over the world. Important and interesting applications can be found in business intelligence, forensics, psychology, and security. A linguistic analysis of a given text can help to identify certain characteristics of the author. For example, companies would like to know, based on the analysis of online product reviews, the demographics of people who like or dislike their products.

In this paper, we describe the participation of our teams (yigal18 and miller18, both teams contain the same people, but in another order) in the PAN 2018 shared task on author profiling. More specifically, the shared task is on gender identification of authors from their tweets and images (100 tweets and 10 images are provided for each author). The addressed languages are English, Spanish and Arabic.

We consider application of several supervised machine learning (ML) methods and various types of features for the gender identification. Content-based features and style-based features are extracted from the tweets and visual features are extracted from the images.

The rest of the paper is organized as follows. Section 2 provides background and presents some related work on text classification in general and author profiling in particular. Section 3 introduces the feature sets (content-based, style-based, and visual) that we have implemented and used in our extensive experiments. Section 4 presents the experimental setup, the experimental results using three modalities (textual feature and / or visual features) for corpora written in three languages and their analysis. Finally, Section 5 summarizes and suggests ideas for future research.

2 Related Work

2.1 Text classification

TC is the supervised learning task of assigning natural language text documents to one or more predefined categories [31]. There are two main types of TC: topic-based classification and style-based classification. An example of a topic-based classification application is classifying news articles to various categories such as Business-Finance, Lifestyle-Leisure, Science-Technology and Sports, that were downloaded from three well-known news web-sites (BBC, Reuters, and TheGuardian) [28].

An example of a style-based classification application is classification based on different literary genres, e.g., action, comedy, crime, fantasy, historical, political, saga, and science fiction [25, 12].

These two classification types often require different types of feature sets to achieve the best performance. Topic-based classification is typically performed using word uni-grams and/or n-grams (n > 2) ([2, 15]. Style-based classification is typically performed

using linguistic features such as quantitative features, orthographic features, part of speech (POS) tags, function words, and vocabulary richness features [25, 16, 17,12].

2.2 Author profiling

Author profiling can be viewed as a sub-task of TC. The author profiling task is of growing importance during recent years. Various author profiling applications are found in business intelligence, forensics, psychology, and security systems. The general aim of an author profiling task is to determine various |demographic information about the text's author(s), e.g., age, cultural background, gender, native language and/or dialect, and various personality traits. In this paper, we will limit ourselves to gender studies because the PAN 2018 shared task is on gender identification.

The gender classification might be viewed as one of the simplest classification tasks. The classification decision can be binary and a relatively large amount of data can be collected. However, such a classification system can be effective only if the writing style between genders does differ [10] and if such stylistic differences can be detected [23].

In contrast to other demographic traits, the link between gender and word use has been extensively studied [34]. Differences in women's and men's language have received relatively high attention within the scientific community as well as in the popular media. However, early studies on gender classification, mainly on formal texts and blogs, reported on accuracies around 75%-80% in most cases [26, 1, 44, 19, 7].

Lakoff [27] found that women use less assertive speech that manifests itself in a higher degree of politeness, less swearing, more frequent tag questions (e.g., "it is …, isn't it?"), more intensifiers (e.g., really, so), and more hedges (e.g., sort of, perhaps, maybe; also known as qualifiers or uncertainty words).

Mulac et al. [32] summarized the findings of more than 30 empirical studies and reported that typical female language features contain intensive adverbs (e.g., really, so), references to emotions, uncertainty verbs (seems to, maybe), negations (e.g., not, never), and hedges; while typical male language features include references to quantity, judgmental adjectives (e.g., good, dumb), elliptical sentences ("Great picture."), directives ("Write that down."), and "I" references.

Schler et al. [45] found that men's writing is more related to money, job and TV, while women's writing is more related to family, sex and eating. Argamon et al. [3]'s experimental results regarding gender showed that the style-based features that proved to be most useful for gender discrimination are determiners and prepositions (markers of male writing) and pronouns (markers of female writing). The content features that prove to be most useful for gender discrimination are words related to technology (male) and words related to personal life or relationships (female).

Pennebaker [35] found that women tend to use more personal pronouns and words referring to emotions. By the contrary, men tend to use more articles, long words (defined as words with more than 6 characters), nouns, and prepositions.

Rangel et al. [39] presented in their overview paper the framework and the results for the Author Profiling task at PAN 2015, which dealt with the identification of age, gender, and personality traits of Twitter users. In comparison to previous years of PAN

[37, 38] the PAN-15 systems achieved significantly higher accuracy values for gender identification. This may suggest that, irrespective the shorter length of individual tweets and their informality, the number of tweets per author is sufficient to profile age and gender with high accuracy. Regarding the features, it was not clear which ones were the most important ones, because the high number of different ones used and combined by the teams. Some of the best teams used the Second Order Representation and other were based on n-gram representation.

A similar phenomenon occurred in the gender classification tasks in [41]. It was difficult to highlight the contribution of any particular feature since the teams used many of them. The second order representation was used by teams that achieved first positions in some of the tasks. Likewise, the distributed representations achieved the first position in gender identification on the Dutch final evaluation.

The best resulting approaches that took part in the gender classification tasks in PAN 2017 [42] took advantage from combinations of n-grams, other content-based features, and style-based features. The best final average gender ranking (for English, Portuguese, and Spanish) shows that the best overall result (82.53%) has been obtained by Basile et al. [4], who used the scikit-learn LinearSVM implementation trained with combinations of character 3- to 5-grams and word 1- to 2-grams with TF-IDF weighting with sublinear term frequency scaling.

3 Features

In this section, we present the various types of features that we applied for the gender identification. Content-based features and style-based features are extracted from the tweets and the visual features are extracted from the images.

3.1 Textual feature sets

In this sub-section, we describe the basic super-sets and sets of the textual features we use for our classification experiments: content-based features and style-based features.

Our content-based features include various n-grams sets, where each one of them is defined by the following template: *number_k-n_type* where number is the number of the features in the set, k is the size of the wanted skip (0 – no skip, 1 – skip of one unit, 2 – skip of 2 units, …), n is code of the grams (1 for unigrams 2 for bigrams, 3 for trigrams, …), and type is W for words or C for characters. All values are represented by TF-IDF values. The specific various n-grams sets that were applied will be presented later in the framework of the experiments.

Our style-based features include the following feature sets: Quantitative features, Orthographic features, Gender features that contain letters or words based on the special tweet text features that appear in Figure 7 in Burger et al. [7], and Gender features that do not contain any letter. The Quantitative set includes 3 features: number of characters in the tweet, number of words in the tweet, and average number of characters in a word. The Orthographic set includes only one feature - the total frequency of the following characters ("/ \ (- : # @ $ & . , ? !") normalized by the number of characters in the tweet. The Gender features that do not contain any letter include the frequency of each

following character strings: ,(: ,_! ,:_ ,!y!, !_i, _i, ooo, !_i, <3, :)$, :(, _:)$, !$, _<3, _b, y:_) normalized by the number of characters in the tweet.

3.2 Visual feature sets

In this sub-section, we describe the three basic sets of visual features we used for our classification experiments: SIFT[1], Color[2], and VGG[3].

The SIFT (Scale Invariant Feature Transform) algorithm was developed by Lowe [29, 30]. The SIFT algorithm detects and describes local features in images. It consists of two sub-processes. The first sub-process is a process that detects interest points in an image. Interest points are where the signal in 2D space has variation that exceeds some threshold criterion and is superior to simple edge detection. The second sub-process creates a vector like descriptor. To create scale invariance, the interest points are scanned at a wide range of scales. For our experiments, we extract from each image at least 500 key-points, and describe each interest point by 128 features. In addition, we create by K-means algorithm a bag of 1000 visual words. For each image, we create a histogram of those words.

The Color algorithm extracts the colors in an image (by the RGB model) and for each color, indicates its frequency (pixels wise) in the image. Using the Colorgram[4] library we extract the 500 most frequent colors in the images, and for each image we build a histogram of the colors in it.

VGG[5] (Oxford Visual Geometry Group) is a group of researchers who built a model containing deep convolutional networks for large-scale image recognition. Using the ORB[6] (Oriented FAST and Rotated BRIEF) system, we extract from each image at least 500 key-points. Then, we describe each key-point by a vector of VGG features. For each user we have ten images; therefore, we have at least 5000 VGG vectors for each user. Then we create for each user a histogram containing 1000 visual words.

4 Experimental Setup and Results

The PAN CLEF 2018 [47] launched an evaluation campaign. Twenty-three teams have participated in this campaign. Each team has proposed its algorithm, which has been evaluated using the TIRA platform [36]. The algorithms and the results of the participated teams have been overviewed in Rangel et al. [43].

General approach: Our approach to authorship profiling is to apply supervised ML methods to TC as was suggested by Sebastiani [44]. The process is as follows. First, given a corpus of training documents, where each document is labeled as either 'male'

[1] https://github.com/abidrahmank/OpenCV2-Python Tutorials/blob/master/source/py_tutorials/py_feature2d/py_sift_intro/py_sift_intro.rst

[2] https://github.com/obskyr/colorgram.py/blob/master/colorgram/colorgram.py

[3] https://docs.opencv.org/trunk/d6/d00/classcv_1_1xfeatures2d_1_1VGG.html

[4] https://pypi.org/project/colorgram.py/#description

[5] https://arxiv.org/abs/1409.1556

[6] http://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_feature2d/py_orb/py_orb.html

or ‘female’, we processed each document to produce values for various combinations of features from different types of features sets: content-based features, style-based features, and visual features. Second, we applied several popular ML methods on the generated combinations of features. Third, we tried additional combinations of features and parameter tuning. Finally, the best model(s) for the training data were tested on out-of-training data (i.e., test data).

Preprocessing: There is a widespread variety of text preprocessing types such as: conversion of uppercase letters into lowercase letters, html object removal, stopword removal, punctuation mark removal, reduction of different sets of emoticon labels to a reduced set of wildcard characters, replacement of HTTP links to wildcard characters, word stemming, word lemmatization, correction of common misspelled words, and reduction of replicated characters. Not all of them are considered as effective by all TC researchers. Many systems use only a small number of simple preprocessing types (e.g., conversion of all the uppercase letters into lowercase letters and / or stopword removal).

In our classification experiments, we tried the following text preprocessing types: L – converting uppercase letters into lowercase letters, U – URL link removal, P – punctuation mark removal, S – stopword removal, C – Error Correction, and T – stemming. The application of the S preprocessing type deletes all instances of 423 stopwords for English text (421 stopwords from Fox [11] plus the letters “x” and “z” that are not found in Fox [11], yet are included in many other stopword lists).

ML methods: We applied four ML methods: MLP– Multilayer Perceptron[7], LinearSVC – SVM with a linear kernel[8], LR - Logistic regression[9], and RF - Random Forest[10].

A brief description of these ML methods follows: MLP is a feedforward neural network ML method [21] where artificial neural network (ANN) can be viewed as a weighted directed graph in which nodes are artificial neurons and directed edges (with weights) are connections from the outputs of neurons to the inputs of neurons. Support vector machine (SVM, also called support vector network) [9] is a model that assigns examples to one of two categories, making it a non-probabilistic binary linear classifier. LinearSVC is SVM with a linear kernel, which is recommended for TC because most of TC problems are linearly separable [22] and training a SVM with a linear kernel is faster compared to other kernels. Logistic regression (LR) is a variant of a statistical model that tries to predict the outcome of a categorical dependent variable (i.e., a class label) [8, 20]. Random Forest (RF) is an ensemble learning method for classification and regression [6]. RF operates by constructing a multitude of decision trees at training time and outputting classification for the case at hand. RF combines the “bagging” idea presented by Breiman [5] and random selection of features introduced by Ho [18] to construct a forest of decision trees.

Tools and information sources: We used the following tools:

- Scikit-learn[11] - a library for ML methods

[7] http://scikit-learn.org/stable/modules/neural_networks_supervised.html

[8] http://scikit-learn.org/stable/modules/generated/sklearn.svm.LinearSVC.html

[9] http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html

[10] http://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html

[11] http://scikit-learn.org/stable/index.html

- NLTK[12]- a library that produces the various n-gram features
- Numpy[13] - a library that performs fast mathematical processing
- Autocorrect[14]- a library that automatically corrects spelling errors.

4.1 Experimental results using the textual feature sets

The first TC experiments were performed for the English corpus using the four ML methods without parameter tuning and any normalization types using only sets of word unigrams. The results are shown in Table 1. The best result is bolded.

Table 1. English corpus - Accuracy results using sets of word unigrams.

Features	MLP	LinearSVC	LR	RF
1000 Word Unigrams	73.06061	75.55556	73.13131	65.77778
2000 Word Unigrams	73.93939	75.65657	73.53535	62.0202
3000 Word Unigrams	74.94949	75.65657	73.53535	64.34343
4000 Word Unigrams	75.85859	76.56566	74.34343	65.15152
5000 Word Unigrams	76.26263	75.85859	74.0404	63.53535
6000 Word Unigrams	76.56566	75.85859	73.43434	66.56566
7000 Word Unigrams	76.56566	75.45455	73.83838	65.45455
8000 Word Unigrams	75.85859	76.26263	73.73737	63.53535
9000 Word Unigrams	**78.28283**	76.76768	73.53535	63.53535
10000 Word Unigrams	77.17172	76.36364	73.63636	65.55556
20000 Word Unigrams	76.46465	76.06061	73.73737	62.72727
30000 Word Unigrams	76.46465	75.9596	73.73737	65.65657
40000 Word Unigrams	75.55556	75.85859	73.53535	62.12121
50000 Word Unigrams	76.26263	76.16162	73.73737	66.56566

Similar experiments have been performed on various sets of word bigrams and the best accuracy result 74.95 was obtained using 20,000 word bigrams.

In our classification experiments, we tried various combinations of text preprocessing types: L – converting uppercase letters into lowercase letters, U – URL link removal, P – punctuation mark removal, S – stopword removal, C – Error Correction, and T – stemming. Only the lowercase preprocessing improved the TC results. Therefore, in our final experiments, we applied only this preprocessing type.

We have also tried various sets of skip character/word n-grams and various style-based feature sets (Quantitative features, Orthographic features, Gender features that contain letters or words, and Gender features that do not contain any letter) in our feature combinations. However, they did not improve the accuracy results.

[12] https://www.nltk.org/

[13] http://www.numpy.org/

[14] https://github.com/phatpiglet/autocorrect

Furthermore, we also applied Principal component analysis (PCA) [24]. However, it did not improve the accuracy results.

Similar results have been achieved for the corpora in Spanish and Arabic. Eventually, based on the results of the previous experiments, we chose to apply the MLP method with the following features for the three languages:

- English: combination of 9,000 word unigrams with the L normalization
- Spanish: combination of 10,000 word unigrams and one orthographic feature with the L normalization for
- Arabic: combination of 7,000 word unigrams and one orthographic feature with the L normalization

This setting was used both by yigal18 and miller18. That is to say, yigal18 and miller18 used the same setting for the experiments on the test data for the textual features.

4.2 Experimental results using the visual feature sets

In the classification experiments using the visual features under the framework of scikit-learn[15] [33], we apply seven versions of five ML methods: (1) Decision tree: default, using mean scaling16; (2) Logistic regression (C=1e-4, max_iter=1e5, using mean scaling); (3) SVM ('poly' kernel): C=1e-3, gamma=.01, using mean scaling; (4) SVM ('rbf' kernel): C=1e-4, gamma=.01; (5) SDGClassifier: epsilon=0.7, max_iter=10000;(6) NN Classifier (version 1): activation='relu', alpha=1e2, hideen_layers=(64,), epsilon=0.8, using mean scaling; and (7) NN Classifier (version 2): activation='relu', epoches=30, hideen_layers=(16,1,).

We used the following sets of visual features: SIFT, SIFT & Color, VGG, VGG & SIFT& Colors, VGG & Colors, VGG & SIFT. Table 2 presents the final results of all possible feature sets combinations (including the basic features sets alone) using the seven versions of five ML methods on the biggest set (2000 training samples, 1000 testing samples). The three best results are in bold.

[15] http://scikit-learn.org/stable/index.html

[16] http://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html

Table 2. The final results of all combinations of visual feature sets using the ML methods.

Feat. Comb. / ML method	SIFT	Color	VGG	SIFT & Color	SIFT & VGG	Color & VGG	SIFT & Color & VGG
Decision Tree	48.58	50.90	50.79	50.60	51.31	52.81	53.23
Logistic Regression	58.99	54.14	61.06	58.48	60.99	58.52	61.54
SGCClassifier	54.75	53.73	56.21	52.12	56.27	55.10	58.00
SVM ('poly' kernel)	60.2	51.33	59.65	58.99	60.16	57.63	60.14
SVM ('rbf' kernel)	48.99	56.50	46.25	48.99	49.03	48.59	47.22
NN Classifier (Sklearn)	57.37	53.23	58.88	58.28	58.18	57.11	60.29
NN Classifier (Keras)	56.77	53.83	61.13	57.78	51.12	56.59	59.69

The best accuracy result in Table 2 is 61.54%, which was obtained by the LR method using all the features (SIFT & Color & VGG). LR obtained the first place for three feature combinations (columns) and SVM ('poly' kernel) obtained the first place for two other feature combinations. The best basic feature set is the VGG (best result in 4 out 7 ML methods). Two ML methods (LR and NN Classifier (Keras)) obtained results above 61% using the VGG features that are very close to the best result in the table that was obtained by all the features. This means that the VGG probably contributes the most and the two other feature sets almost did not contribute.

For the first model (yigal18) we used the following features: 1000 VGG features, 1000 SIFT features, and 500 Color features. The learning method was a Logistic Regression ($C=1e^{-2}$), with mean scaling.

For the second model (miller18), we used only the 1000 VGG features. The chosen ML method was the NN Classifier (Keras) (with 64 hidden units at one layer).

4.3 Experimental results using both the textual and visual feature sets

For this experiment, we used two different models (yigal18 and miller18). The construction of each model was performed according to the same general method: the data was separately classified by the visual and textual features, and using the next formula we combined the two classifications into a final classification.

$$CombainedProb(class_i) = \alpha * P_{textual}(class_i) + (1-\alpha) * P_{visual}(class_i)$$

For each model, we used the same setting for the textual features (see sub-section 4.1) and the MLP method. Different settings of the visual features and different ML methods have been applied by the two models.

The first model (yigal18) applied Logistic Regression on the following visual features: 1000 VGG features, 1000 SIFT features, and 500 Color features. The value of α was set to 0.5, which means that we gave equal weighs for the textual and visual components.

The second model (miller18) used as its visual features only the 1000 VGG features. The chosen ML method was the NN Classifier (Keras) (with 64 hidden units at one layer). The textual features remain the same since their results in the first model were relatively high. The value of α was set to 0.75 (i.e., giving higher weight for textual features because they obtained better results).

The results of the two models yigal18 and miller18 are presented in Tables 3 and 4, respectively. The best results for the other modalities are in bold.

Table 3. The accuracy results (in %) of the first model (yigal18).

	Textual	Visual	Combined
English	79.11	49.42	78.89
Arabic	75.9	**51**	**75.7**
Spanish	76.5	**50.27**	75.91

Table 4. The accuracy results (in %) of the second model (miller18) using the MLP method.

	Textual	Visual	Combined
English	79.11	**51.74**	**79.47**
Arabic	75.9	49.7	**75.7**
Spanish	76.5	49.23	**76.23**

The comparison of the yigal18 (model1, Table 3) and miller18 (model2, Table 4) leads to the following findings and perhaps to the following conclusions. (1) Regarding the visual results, there is a decrease in two languages (Spanish & Arabic) and an increase in English. A possible explanation is that using all the visual features (SIFT & Color& VGG) leads to slightly better results than using only the VGG features. (2) Regarding the combined results, there is an increase in two languages (English & Spanish) and no change for Arabic. The improvements in miller18 (model2) are probably because of the upgrade of alpha from 0.5 to 0.75. That is to say, i.e., higher weight for the textual features leads to better combined results. (3) The second visual model run faster due to the lower number of features (1000 VGG features), in contrast to the first model (all the 2500 visual features). (4) The result of the combined features for model2 is better than the result of the textual features for model2. This finding means that the visual features slightly contribute to improve the result obtained by the textual features. This phenomenon did not occur in all other cases in both models.

5 Summary and Future Work

In this paper, we describe our participation in the PAN 2018 shared task on author profiling, identifying authors' gender, based on their images and tweets. We tried

various pre-processing types, a widespread variety of feature sets, and four ML methods.

The best results using the textual features were achieved using the MLP method after applying the L normalization and using 9,000 word unigrams for English, 10,000 word unigrams and one orthographic feature for Spanish, and 7,000 word unigrams and one orthographic feature for Arabic.

The best result (61.54%) for the visual features was obtained by the LR method using all the features (SIFT & Color & VGG). The best basic feature set is the VGG. Two other results (also above 61%) were obtained while applying LR and NN (Keras) using the VGG features. This means that the VGG probably contributes the most and the two other feature sets almost did not contribute.

The best results using the combined features were achieved using model2 with 0.75 as a weight to the textual features and 0.25 as a weight to the visual features. For the textual features we used the best model that was used also in model1 (sub-section 5.3). For the visual features of the second model, we use only the 1000 VGG features. The chosen ML method was the NN Classifier (Keras) (with 64 hidden units at one layer).

Future research proposals include: (1) applying additional combinations of feature sets; (2) tuning each model (textual and visual) separately; (3) tuning the value of alpha to find the best combined model; (4) developing a new combined model performing multi-modal fusion using the textual and visual modalities; (5) applying various deep neural models; and (6) building model that will perform author profiling in general and gender prediction in particular using keyphrases [13,14] that distinguish each of classes in general and each gender in paticular.

Acknowledgments. This work was partially funded by the Jerusalem College of Technology (Lev Academic Center) and we gratefully acknowledge its support.

References

1. Argamon, S., Koppel, M., Fine, J., Shimoni, A. R.: Gender, genre, and writing style in formal written texts. TEXT-THE HAGUE THEN AMSTERDAM THEN BERLIN-, 23(3), 321-346 (2003).
2. Argamon, S., Whitelaw, C., Chase, P., Hota, S. R., Garg, N., Levitan, S.: Stylistic text classification using functional lexical features: Research articles. Journal of the American Society for Information Science and Technology. 58, 6, 802–822 (2007).
3. Argamon, S., Koppel, M., Pennebaker, J. W., Schler, J.: Automatically profiling the author of an anonymous text. Communications of the ACM, 52(2), 119-123 (2009).
4. Basile, A., Dwyer, G., Medvedeva, M., Rawee, J., Haagsma, H., Nissim, M.: N-GrAM: New Groningen Author-profiling Model. arXiv preprint arXiv:1707.03764 (2017).
5. Breiman, L.: Bagging predictors. Univ. California Technical Report No. 421. (1994).
6. Breiman, L.: Random forests. Machine learning, 45(1), 5-32 (2001).
7. Burger, J. D., Henderson, J., Kim, G., Zarrella, G.: Discriminating gender on Twitter. In Proceedings of the conference on empirical methods in natural language processing (pp. 1301-1309). Association for Computational Linguistics (2011).

8. Cessie, S. Le, Van Houwelingen, J. C.: Ridge estimators in logistic regression, Applied statistics, pp. 191-201 (1992).
9. Cortes, C., Vapnik, V.: Support-vector networks. Machine learning, 20(3), 273-297 (1995).
10. Eckert, P., McConnell-Ginet, S.: Language and gender. Cambridge University Press (2013).
11. Fox, C.: A stop list for general text. In Acm sigir forum (Vol. 24, No. 1-2, pp. 19-21). ACM (1989).
12. Gianfortoni, P., Adamson, D., Rosé, C. P.: Modeling of stylistic variation in social media with stretchy patterns. In Proceedings of the First Workshop on Algorithms and Resources for Modelling of Dialects and Language Varieties (pp. 49-59). Association for Computational Linguistics (2011).
13. HaCohen-Kerner, Y., Gross, Z., Masa, A.: Automatic extraction and learning of keyphrases from scientific articles. In Int. Conf. on Intelligent Text Processing and Computational Linguistics, Springer, Berlin, Heidelberg, pp. 657-669 (2005).
14. HaCohen-Kerner, Y., Stern, I., Korkus, D., Fredj, E.: Automatic machine learning of keyphrase extraction from short html documents written in Hebrew. Cybernetics and Systems: An International Journal, 38(1), 1-21 (2007).
15. HaCohen-Kerner, Y., Mughaz, D., Beck, H., Yehudai, E.: Words as classifiers of documents according to their historical period and the ethnic origin of their authors. Cybernetics and Systems: An International Journal, 39(3), 213-228 (2008).
16. HaCohen-Kerner, Y., Beck, H., Yehudai, E., Rosenstein, M., Mughaz, D.: Cuisine: Classification using stylistic feature sets &/or name-based feature sets. Journal of the American Society for Information Science and Technology 61 (8), 1644–57 (2010A).
17. HaCohen-Kerner, Y., Beck, H., Yehudai, E., Mughaz, D.: Stylistic feature sets as classifiers of documents according to their historical period and ethnic origin. Applied Artificial Intelligence 24 (9), 847–62 (2010B).
18. Ho, T. K.: Random Decision Forests. Proceedings of the 3rd Int. Conference on Document Analysis and Recognition, Montreal, QC, 14–16 August 1995. 278–282 (1995).
19. Holmes, J., Meyerhoff, M. (Eds.).: The handbook of language and gender (Vol. 25). John Wiley & Sons (2008).
20. Hosmer Jr, D. W., Lemeshow, S., Sturdivant, R. X.: Applied logistic regression (Vol. 398). John Wiley & Sons (2013).
21. Jain, A. K., Mao, J., Mohiuddin, K. M.: Artificial neural networks: A tutorial. Computer, 29(3), 31-44 (1996).
22. Joachims, T.: Text categorization with support vector machines: Learning with many relevant features. In European conference on machine learning (pp. 137-142). Springer, Berlin, Heidelberg (1998).
23. Jockers, M. L., Witten, D. M.: A comparative study of machine learning methods for authorship attribution. Literary and Linguistic Computing, 25(2), 215-223 (2010).
24. Jolliffe, I. T.: Principal component analysis and factor analysis. In Principal component analysis (pp. 115-128). Springer, New York, NY (1986).
25. Kessler, Brett, Nunberg, Geoffrey, Hinrich Schutze.: Automatic detection of text genre. In P. R. Cohen & W. Wahlster (Eds.), In Proc. of the 35th annual meeting of the

ACL and 8th conf. of the European chap. of the Assoc. for Computational Linguistics. 32-38, Somerset, New Jersey: Association for Computational Linguistics (1997).
26. Koppel, M., Argamon, S., Shimoni, A. R.: Automatically categorizing written texts by author gender. Literary and linguistic computing, 17(4), 401-412 (2002).
27. Lakoff, R.: Language and woman's place. Language in society, 2(1), 45-79 (1973).
28. Liparas, D., HaCohen-Kerner, Y., Moumtzidou, A., Vrochidis, S., Kompatsiaris, I. News articles classification using Random Forests and weighted multimodal features. In Information Retrieval Facility Conference, Springer, Cham, pp. 63-75, (2014).
29. Lowe, David G.: Object recognition from local scale-invariant features. In Proc. of the Int. Conf. on Computer Vision. 2., pp. 1150–1157 (1999).
30. Lowe, David G.: Distinctive image features from scale-invariant keypoints. International journal of computer vision, 60(2), 91-110 (2004).
31. Meretakis, D., Wuthrich, B.: Extending naive Bayes classifiers using long itemsets, Proc. of the 5th ACM-SIGKDD Int. Conf. Knowledge Discovery, Data Mining (KDD'99), San Diego, USA, 165-174 (1999).
32. Mulac, A., Bradac, J. J., Gibbons, P.: Empirical support for the gender-as-culture hypothesis: An intercultural analysis of male/female language differences. Human Communication Research, 27(1), 121-152 (2001).
33. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... Vanderplas, J.: Scikit-learn: Machine learning in Python. Journal of machine learning research, 12(Oct), 2825-2830 (2011).
34. Pennebaker, J. W., Mehl, M. R., Niederhoffer, K. G.: Psychological aspects of natural language use: Our words, our selves. Annual review of psychology, 54(1), 547-577 (2003).
35. Pennebaker, J.W.: The secret life of pronouns: What our words say about us. New York: Bloomsbury Press (2011).
36. Potthast, M., Gollub, T., Rangel, F., Rosso, P., Stamatatos, E., Stein, B.: Improving the Reproducibility of PAN's Shared Tasks: Plagiarism Detection, Author Identification, and Author Profiling. In: Kanoulas, E., Lupu, M., Clough, P., Sanderson, M., Hall, M., Hanbury, A., Toms, E. (eds.) Information Access Evaluation meets Multilinguality, Multimodality, and Visualization. 5th International Conference of the CLEF Initiative (CLEF 14). pp. 268–299. Springer, Berlin Heidelberg New York (Sep 2014).
37. Rangel, F., Rosso, P., Koppel, M., Stamatatos, E., Inches, G.: Overview of the author profiling task at PAN 2013. In CLEF Conference on Multilingual and Multimodal Information Access Evaluation, CELCT, pp. 352-365 (2013).
38. Rangel, F., Rosso, P., Chugur, I., Potthast, M., Trenkmann, M., Stein, B., ... Daelemans, W.: Overview of the 2nd author profiling task at pan 2014. In CLEF 2014 Evaluation Labs and Workshop Working Notes Papers, Sheffield, UK, 2014, pp. 1-30 (2014).
39. Rangel, F., Rosso, P., Potthast, M., Stein, B., Daelemans, W.: Overview of the 3rd Author Profiling Task at PAN 2015. In CLEF p. 2015 (2015).
40. Rangel, F., Franco-Salvador, M., Rosso, P.: A low dimensionality representation for language variety identification. In International Conference on Intelligent Text Processing and Computational Linguistics Springer, Cham, pp. 156-169 (2016A).

41. 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 Working Notes Papers of the CLEF 2016 Evaluation Labs. CEUR Workshop Proceedings/Balog, Krisztian [edit.]; et al., pp. 750-784 (2016).
42. 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. Working Notes Papers of the CLEF (2017).
43. Rangel, F., Rosso, P., Montes-y-Gómez, M., Potthast, M., Stein, B.: Overview of the 6th Author Profiling Task at PAN 2018: Multimodal Gender Identification in Twitter. In: Cappellato, L., Ferro, N., Nie, J.Y., Soulier, L. (eds.) Working Notes Papers of the CLEF 2018 Evaluation Labs. CEUR Workshop Proceedings, CLEF and CEUR-WS.org (Sep 2018)
44. Sebastiani, F.: Machine learning in automated text categorization. ACM computing surveys (CSUR), 34(1), 1-47 (2002).
45. Schler, J., Koppel, M., Argamon, S., Pennebaker, J. W.: Effects of age and gender on blogging. In AAAI spring symposium: Computational approaches to analyzing weblogs, Vol. 6, pp. 199-205 (2006).
46. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014).
47. Stamatatos, E., Rangel, F., Tschuggnall, M., Kestemont, M., Rosso, P., Stein, B., Potthast, M.: Overview of PAN-2018: Author Identification, Author Profiling, and Author Obfuscation. In: Bellot, P., Trabelsi, C., Mothe, J., Murtagh, F., Nie, J., Soulier, L., Sanjuan, E., Cappellato, L., Ferro, N. (eds.) Experimental IR Meets Multilinguality, Multimodality, and Interaction. 9th Int. Conference of the CLEF Initiative (CLEF 18). Springer, Berlin Heidelberg New York (Sep 2018).