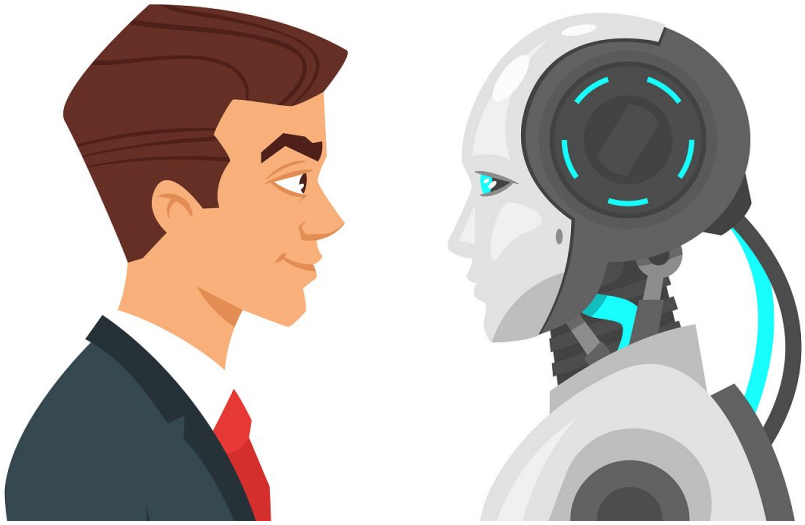
Bot Detection and Gender Profiling

Andrea Bacciu
Student ID: 1747105



Contents

1 Author Profiling - Introduction

Author profiling is a technique used to create a user profile, i.e. an analysis of the author of a text. The information that can be extracted from the text is manifold such as age, gender, native language and personality traits. Author profiling is used by companies to improve the quality of the service, offering personalized suggestions and experiences to users to maximize customer satisfaction and profits. It can be applied for forensic activities where the authors' profile is used as additional evidence in criminal investigations and security. Knowing the demographics of social media users (age and gender), cultural and social context, mother tongue and dialects, can help identify potential terrorists.

A known use of bots is to influence public opinion by hiding among real users. In fact, is known is the case of the 2016 presidential election in the United States where public opinion towards both parties (D. Trump and H. Clinton) seems to have been influenced by the massive amount of bots present on the Twitter platform.

2 PAN

PAN is a series of scientific events and shared tasks (similar to the competition site called Kaggle) on digital text forensics and stylometry. In each edition, the organizers propose several challenges related to the forensic analysis, provide the guidelines and the dataset to accomplish the tasks. Participants team can concur for one or all of them.

3 Project idea

The project aims to identify the nature of the population of the well-known social network Twitter. To do this I worked in the development of different models in two different languages namely English and Spanish. The model inputs exactly one hundred tweets from each account and classifies it at first into two bot or human classes. All accounts classified as belonging to the human class will form the test set of the second

phase. The second step is to reclassify all accounts previously classified as human into two additional male or female classes.

4 Dataset and Metrics

The dataset was provided by the PAN organization. The dataset is made up by two sets of Twitter accounts, one for the English language, and the other for the Spanish. Each account in the dataset is a collection of one hundred tweets.

Constraints in the dataset:
The dataset has some constraints: each author has his own username obscured by a hash. It is not possible to trace through which device the tweets were made. The date, time of publication and the profile picture of the account are not available, all the links present have been normalized along a single internet domain.

The dataset was already split into two parts train and dev and correlated with the labels. In the table 1 are reported some statistics about the dataset.

Table 1: Train Dataset Composition

	English				Spanish			
	Bots	Female	Male	Total	Bots	Female	Male	Total
Train	1440	720	720	2880	1040	520	520	2080
Dev	620	310	310	1240	460	240	240	920
Total	2060	1030	1030	4120	1380	760	760	3000

In order to draw up a ranking, PAN developed two private test sets, the specifications of which were not made available until the end of the task. We will call these two private tests test1 and test2 respectively, as shown in the table 2.

Table 2: Private test

	English				Spanish			
	Bots	Female	Male	Total	Bots	Female	Male	Total
Test1	132	66	66	264	90	45	45	180
Test2	1320	660	660	2640	900	450	450	1800

4.1 Metrics

The metric used for the task is accuracy. This is because as you can see in the table the classes are balanced. There are human K and bot for the first phase, while for the

second phase we have $K/2$ men and $K/2$ women.

$$accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{1}$$

5 Bot Detection

5.1 Pre-processing

To manage the dataset I performed many preprocessing techniques, sometimes I used different techniques depending on the features extracted.

5.1.1 Tokenization

As a first technique I used the tokenization of the text. I used the **TweetTokenizer** , this tokenizer is designed to be flexible and easy to adapt to tweets. It can handle out of the box ASCII emoticons, and it replaces character sequences of length greater than 3 with sequences of length 3.

Table 3: Tweet Tokenizer

Original Text	**Tweet Tokenizer**
This is a cooool #dummysmiley: :-) :-P < 3 and some arrows $<> - > < --$	'This', 'is', 'a', 'cooool', '#dummysmiley', ':', ':-)', ':-P', '< 3', 'and', 'some', 'arrows', '$<$', '$>$', '$- >$', '$< --$'

5.1.2 Text Distortion

The text distortion is a preprocessing techniques taken from the Authorship Attribution field. During my during my work in the research of author profiling papers I have appreciate the work of Stamatatos, in the paper [5], he use the distortion in order to generalize the feature extraction from the text indipendently from the topic (domain) of the text argument and hidding the word and pointing out the style. This function transforms every ASCII characters into * and leaves unchanged no-ASCII characters. An example of distorted text is shown in Tab. 4.

Table 4: Text distortion

Original Text	**Distorted Text**
RT @BIBLE_Retweet: Great men are not always wise - Job 32:9	** @*****_*******: ***** *** *** *** ****** **** - *** **:*
I don't know. Just making conversation with you, Morty. What do you think, I-I-I... know everything about everything?	* ***'* ****. **** ****** ********** **** ***, *****. **** ** *** *****, *-*-*... **** ********** ***** **********?

[0]TweetTokenizer: `https://www.nltk.org/api/nltk.tokenize.html`

5.2 Features

This section describes the features used for bot detection model. For each account we compute the following features:

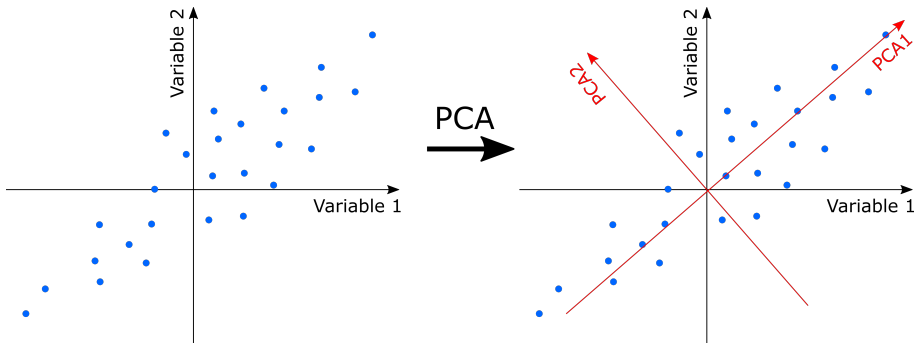
- **Emojis**: The average number of emojis used in each tweet.
- **Web link**: The average number of links shared in each tweet.
- **Hashtag**: The average number of hashtags used.
- **Len of Tweets**: The average length of the tweets.
- **Len of ReTweets**: The average length of the retweets.
- **Semicolons**: The average number of semicolons used.
- **Cosine Similarity score**: For all tweets that belong to the same user, I calculate the weight of each word with the TF-IDF. Then, for each pair of tweets, I compute the cosine-similarity. Finally, I get as feature the average of the scores. The idea is that the cosine similarity of bots is higher than that of humans. Since the messages that belong to the same Bot, tend to be very similar among themselves.
- **Sentiment Analysis**: I perform the sentiment analysis to each tweet, then I use as features the average of the neutral sentiment score and the average on the compound score. To perform the sentiment analysis, I use the Vader Sentiment Analysis tool [3]. It provides for the analyzed sentence a positive, a negative, a neutral and a compound score where the compound score is a single uni-dimensional measure that sums up the sentiment of the sentences.
- **Text Distortion**: In this feature I used the preprocessing technique illustrated previously. I concatenated all the tweets into one text, then I have replaced all the emoticons with the same tag ::EMOJI::, in this way we normalize all the emoticons, and at the same time after the text distortion we can still have a recognizable pattern (::*****::). Finally, I apply the distortion techniques and I have extracted from the text the first 1000 char-grams by frequency of length from 2 up to 8, that appear at least 5 times among all the tweets of the account. The selected char-grams are then weighted with Tf-Idf in which the term frequency is logarithmically scaled.

5.3 Dimensionality Reduction

Reducing the feature dimensionality is a crucial aspect task during the machine learning model design.

- It improves the generalization of the model, reducing the problems of Overfitting.
- Reduction of train times.
- Simplification of the model.

Feature selection:
My first attempt to reduce the dimension of feature vector is the selection of the best features with the SelectKBest algoritmh provided by the scikit-learn framework. For the TF-IDF matrix, I have selected the best 1000 subsequence of char and this action has improved my model in terms of accuracy and speed. At the end of this process, I have a feature vector of 1009 features for english bot task and 1008 for the spanish.
Principal Component Analysis — PCA I have done the scaling of my feature vector using the StandardScaler provided by scikit-learn, then I used the Principal Component Analysis [2, 4] — PCA to reduce dimensionality. The PCA project features from a feature space of N-dimensionality into another one with K-dimensionality (where N > K) with minimal loss of information. For the English bot classifier, I have reduced the dimension of the features from 1009 to 56, while for the Spanish bot classifier from 1008 to 46. In both the cases I have set the *withen* parameter of PCA equals to True.



5.4 Classifier

To address the problem I have experimented with different classification algorithms, such as the use of the excellent random forest, support vector machine, ensable techniques, multi layer percetron.
Spanish: From my experiments I found a better generalization using the Support Vector Machine algorithm with the Radial basis function alias RBF as kernel and all hyperparameter with the default values .
English: In the English case instead, I found that the best performances are obtained through a neural network, composed of 5 hidden layers in which the activation function is the Rectified Linear Units — ReLU has been applied. I applied a regularization technique called Dropout trying to avoid overfitting problems. Each layer is interleaved by a dropout layer applied with the 30% of probability. The output layer instead uses the sigmoid as activation function, being in the case of binary classification. This is because the sigmoid function maps each value into a range $[0, 1]$. As an optimizer I used Adam together with a batch size of 10 examples. Then calculating the loss through the Cross Entropy specifically the binary version. I used 25 epochs for training the model.

5.5 Hyperparameter Optimization

Despite several tests together with the Grid Search algorithm, I found that for the **Spanish** model the best set of hyperparameters is the default (as set in the SKlearn framework) with the Radial basis function alias RBF as kernel. In the **English** model I found the best settings using the KerasClassifier in combination with the GridSearch Algoritmh and cross-validation $= 5$. KerasClassifier is a wrapper used in models created with Keras to be used in the sklearn framework pipeline.

5.6 Results

As you can see in the table 5 in all 3 evaluations, the English model has achieved higher results than the Spanish model. We can also note that that both models generalize well, obtaining similar performances on all three evaluations.

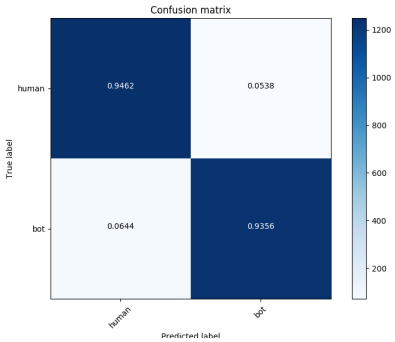


Figure 1: Bot Confusion Matrix - English

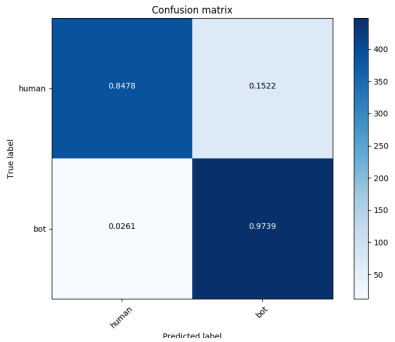
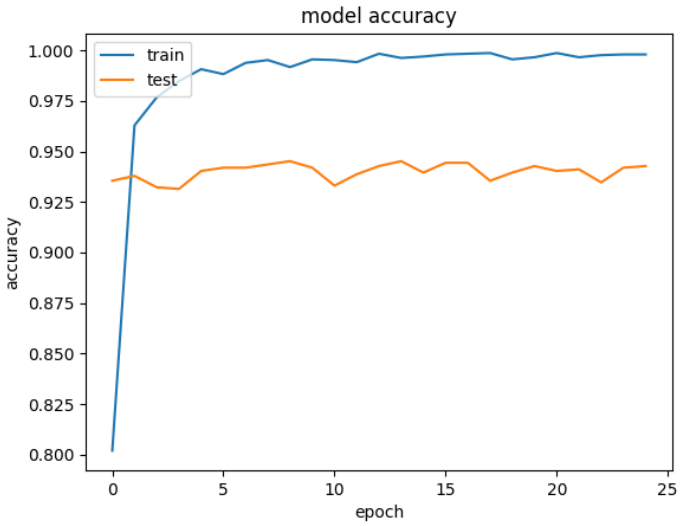


Figure 2: Bot Confusion Matrix - Spanish

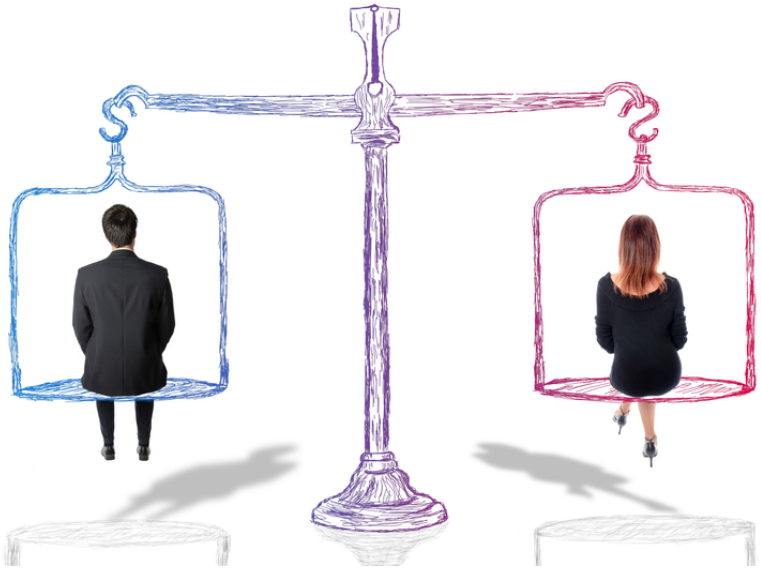
Table 5: Bot detection evaluation results

Test	English	Spanish
Dev	94.46%	92.06%
Test1	93.35%	88.89%
Test2	94.40%	90.78%

Figure 3: English bot classifier accuracy on dev test



6 Gender Profiling



6.1 Pre-processing

This section illustrates the preprocessing techniques used for gender profiling. I have tokenized the text and then I normalized the text through the stemming for English and the lemmatization for Spanish and finally I concatenated all the tweets of an author into a single text.

6.1.1 Tokenization

As previously done for bot detection, I also tokenized the text for gender profiling, always using the **TweetTokenizer**,

6.1.2 Text Normalization

I normalized the text by removing the declinations of the words based on the context, with the aim of better generalizing and capturing the differences between the words chosen between man and woman as suggested in the literature.
Stemming:
I used the Snowball Stemming algoritmh provied by the Natural Language ToolKit (NLTK). The stemming is the process of reducing a word from its inflected form to its root form.
This technique serves to generalize a word, abstracting it from the context in which it is written.
For English I used the stemming while for Spanish I used the lemmatization. This choice was made after numerous tests by measuring their performance through cross-validation.

Table 6: Stemming

Original Text	**Stemmed Text**
He was running and eating at same time. He has bad habit of swimming after playing long hours in the Sun.	He wa run and eat at same time. He ha bad habit of swim after play long hour in the sun.

Lemmatization I used the Lemmatization algoritmh provied by the spaCy framework. Lemmatization, unlike Stemming, reduces inflected words by ensuring that the root of the word belongs to the language.
The root of the word is called Lemma and is the canonical form or form of the dictionary. As with Stemming, this technique serves to generalize a word, abstracting it from the context in which it is written.

6.2 Feature

I have used the same kind of feature, Word n-gram with the TF-IDF algoritmh, but with different settings for English and Spanish.
In the **english** model I selected the 90000 most frequent words n-grams of length from

Table 7: Lemmatizzation

Original Text	**Lemmatized Text**
He was running and eating at same time. He has bad habit of swimming after playing long hours in the Sun.	He wa running and eating at same time. He ha bad habit of swimming after playing long hour in the Sun.

1 up to 5, that have a document frequency higher than 4.
In the **spanish** case, I used words n-grams of length from 1 up to 8, that appear in each document at least 7 times. In the spanish case I selected the 50,000 most frequents n-grams.

6.3 Dimensionality Reduction

In the Gender Profiling task in order to reduce the dimensionality I used the Latent Semantic Analysis. The Latent Semantic Analysis—LSA technique [1] is a statistical approach to extracting relations among words by means of their contexts of use in documents. It makes no use of natural language processing techniques for analyzing morphological, syntactic, or semantic relations, nor does it use humanly constructed resources. LSA starting from the TF-IDF, applies a reduced-rank singular value decomposition (SVD) on the TF-IDF matrix to reduce the number of rows while preserving the similarity structure among columns.

6.4 Classifier

During the experiments I tried different classifiers such as: Logistic Regression, Random Forest and SVM. To evaluate the performance of the classifiers, I tested the performance on a dataset having all humans (over the dev set) not considering the error bias generated by the first phase of the problem.
As can be seen in both cases, the best performances were achieved by a single SVM.

6.5 Hyperparameter Optimization

Spanish: To find the best set of hyperparameters I used the Grid Search algorithm with cross-validation $= 5$. Grid Search allowed me to find the following parameters: In the English model I used a penalty coefficient of 12.000 while for the Spanish of 5.000 with the Radial basis function alias RBF as kernel, the remaining parameters have the default values.

6.6 Results

The table 8 shows the results obtained by the two gender profiling models. Here, too, it is possible to note that the English model always performs better. The table shows two tests on the dev set. The line with the inscription "(No Bot)" shows the results obtained by the model without the error bias generated by the bot detection phase. The other

results in the table show the behavior of the models following the results produced by the bot phase.

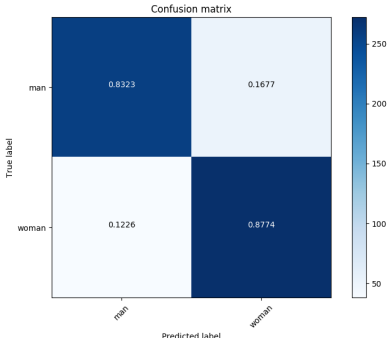


Figure 4: Gender Confusion Matrix - English

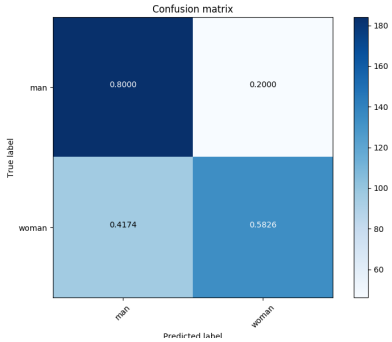
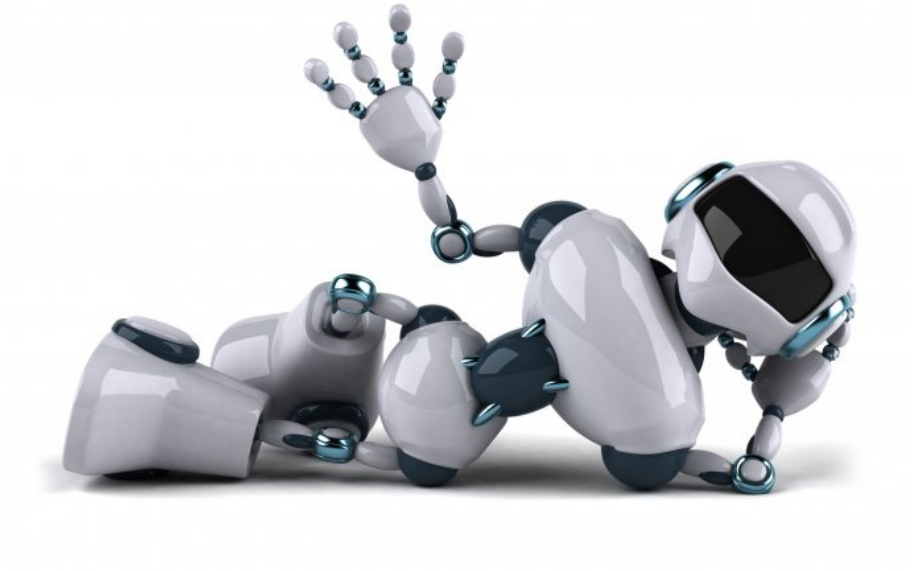


Figure 5: Gender Confusion Matrix - Spanish

Table 8: Final results of Gender Profiling

Test	English	Spanish
Dev (No Bot)	$85,48\%$	$69,13\%$
Dev	$80,42\%$	$63,68\%$
Test1	$80,68\%$	$69,44\%$
Test2	$84,17\%$	$77,61\%$



7 Conclusions

The results highlight several factors. Bot detection and gender profiling achieves higher results in English than Spanish. In the case of English bot detection with a neural network, I obtained about 1% higher performance in terms of accuracy than an SVM, despite this, the Spanish model continues to perform better with the use of the SVM. The distortion technique has made a huge contribution in both bot detection models bringing the accuracy of both models above 80%. In the gender profiling model, on the other hand, I have not benefited from any type of hand crafted feature. I tried to work a lot on using the distortion technique and the use emojis to capture the differences between men and women but I didn't get the desired results. The use of a neural network has been somewhat disappointing compared to the use of Random Forest or SVM. An interesting factor is the importance of techniques such as PCA and LSA in improving performance both in terms of accuracy and in terms of computation.

References

[1] Susan T Dumais. Latent semantic analysis. *Annual review of information science and technology*, 38(1):188–230, 2004.

[2] Harold Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of educational psychology*, 24(6):417, 1933.

[3] Clayton J Hutto and Eric Gilbert. Vader: A parsimonious rule-based model for sentiment analysis of social media text. In *Eighth international AAAI conference on weblogs and social media*, 2014.

[4] Karl Pearson. Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11):559–572, 1901.

[5] Efstathios Stamatatos. Authorship attribution using text distortion. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 1138–1149, 2017.