Journal of Intelligent & Fuzzy Systems xx (20xx) x–xx
DOI:10.3233/JIFS-179894
IOS Press

Author detection: Analyzing tweets by using a Naïve Bayes classifier

Rocío Abascal-Mena* and Erick López-Ornelas
Department of Information Technologies, Universidad Autónoma Metropolitana, Unidad Cuajimalpa. Avenida Vasco de Quiroga 4871, Colonia. Santa Fe, Cuajimalpa. Delegación Cuajimalpa de Morelos. C.P. 05348, Ciudad de México, México

Abstract. In the context of digital social media, where users have multiple ways to obtain information, it is important to have tools to detect the authorship within a corpus supposedly created by a single author. With the tremendous amount of information coming from social networks there is a lot of research concerning author profiling, but there is a lack of research about the authorship identification. In order to detect the author of a group of tweets, a Naïve Bayes classifier is proposed which is an automatic algorithm based on Bayes' theorem. The main objective is to determine if a particular tweet was made by a specific user or not, based on its content. The data used correspond to a simple data set, obtained with the Twitter API, composed of four political accounts accompanied by their username and tweet identifier as it is mixed with multiple user tweets. To describe the performance of the classification model and interpret the obtained results, a confusion matrix is used as it contains values like accuracy, sensitivity, specificity, Kappa measure, the positive predictive and negative predictive value. These results show that the prediction model, after several cases of use, have acceptable values against the observed probabilities.

Keywords: Naïve Bayes classifier, authorship detection, social network analysis, Twitter, confusion matrix

1. Introduction

Social media has changed the way people communicate and receive information about what happens daily. In this way, research areas like computer science evolve to have all the necessary methodological tools in order to understand social, political and economical news as quick as they are required by the modern society. Particularly, social media like Twitter has become the go-to place for latest developments [3].

It is undeniable that as a communication platform, Twitter has increasingly infused itself into daily life. So, one of the greatest tasks for academe is to understand why, how, when and with whom does the society communicates daily. In order to answer the last question, it is important to consider the tremendous amount of information, that occurs in social media, which is viable to be collected, in real time, directly from users; so, data can be analyzed by composing a corpus and applying linguistic research. However, it is difficult to identify the author of a post in order to hierarchize it according to what or who the user wants to read first. Even though it is possible to know the authorship by looking at the username, however in public accounts, like politicians, the reality is that a group of people is behind these posts. It is important to identify in what cases the owner of the account is the same person that is publishing. This

*Corresponding author. Rocío Abascal-Mena, Department of Information Technologies, Universidad Autónoma Metropolitana, Unidad Cuajimalpa. Avenida Vasco de Quiroga 4871, Colonia. Santa Fe, Cuajimalpa. Delegación Cuajimalpa de Morelos. C.P. 05348, Ciudad de México, México. E-mail: mabascal@correo.cua.uam.mx.

can be addressed like a classification problem where given a set of classes the objective is to seek to which of them a given post or tweet, in the case of social networks, belongs to.

Among the most popular machine learning algorithms, for implementing text classification models, the followings are found: Naïve Bayes algorithm's family, Support Vector Machines and deep learning. In this paper, an approach for the detection of the authorship, of a group of tweets, is based on the use of a Naïve Bayes classifier which is an automatic algorithm based on Bayes' theorem. The main objective is to determine if a particular tweet was made by a specific user or not, based on its content. The data used correspond to a simple data set, obtained with the Twitter API, which belong to four political accounts accompanied by their username and tweet identifier as it is mixed with multiple user tweets. To describe the performance of the classification model and interpret the obtained results, a confusion matrix is used as it contains values like sensitivity, specificity, Cohen´s Kappa measure, the positive predictive value and negative predictive value. The utility of using Naïve Bayes to classify authors according to the tweet is demonstrated as the classification model it is improved.

The rest of this paper is organized as following: section 2 gives a focused overview of related work. In section 3, the classification of tweets is explained by following four main steps. Section 4 is dedicated to the case of use while section 5 is devoted to the evaluation and interpretation of the obtained results. Finally, section 6 gives some conclusions about the work and opens several perspectives.

2. Related work

The authorship recognition or author identification is a subfield of Natural Language Processing (NLP) that uses machine learning techniques to be able to identify the author of a text based on characteristics such as vocabulary and frequency of terms. Generally, machine learning techniques have been used in the study of large texts. However, the short length of Twitter messages present a rarely examined challenge that have been also studied in a small amount of recent research. Short length creates new challenges like:

- consist of terse text, Twitter has a limitation of length;
- is an informal way of communication, which lacks syntactical forms of formal writing;
- content is inconsistent and topics may vary over time and
- contain highly unstructured data as a combination of weblinks, hashtags, emoticons, special symbols and images are present along with texts [6].

Concerning Twitter, there is a lot of research work in the analysis of tweets like hashtag recommendations, sentiment analysis, opinion mining and text classification. The classification of text, in this case tweets, is often done according to general themes or categories like health, education or politics. However, there is a lack of studies concerning the veritable identity of the user behind the tweet. Some studies have been oriented to the identification of anonymous users in Twitter, like in [1] where the objective is to uncovered them using only linguistic stylometry.

In [8], the authors examine potential avenues of author identification, in tweets, by using supervised learning methods for data classification as Support Vector Machines (SVM). In this case, Bag-of-Words (BOW) and Style Marker feature sets were extracted and evaluated through a series of experiments where the Style Marker feature sets were found to be significantly more useful than BOW and are therefore suggested for potential applications in future research [8]. The same approach is followed in [6] were a BOW and style-based markers have been used for the identification of tweets authorship obtaining a recognition rate of 97% on a database of 70 Twitter users, which validates the superiority of using social interactive data compared to traditional linguistic profiles.

By using a set of stylistic markers, including personal an idiosyncratic edition options, an automatic authorship analysis was conducted in Twitter messages among three authors [9]. In this study, SVM classifiers where used demonstrating that they can be useful to attribute authorship in combination with a group of content-agnostic features. Some works in analyzing Twitter are focused in the identification of the gender and language of the author, as in [4, 5] where n-grams or BOW model are used. In [7] a variety of algorithms, coming from WEKA, were used as some machine learning classifiers were studied in order to compare their performance in the identification of troll profiles based on their published tweets. In this case, Random Forest, J48, K-Nearest Neighbor (KNN), Sequential Minimal Optimization (SMO) and Bayes Theorem-based algorithms were

implemented resulting that SMO and decision trees were the most appropriated for this case of use.

Classification methods using traditional approaches like BOW have some limitations, so in [2], a multi-label classifier is used based on a small set of domain-specific features extracted from the author's profile and the text in order to classify authors into a predefined set of generic classes [2]. Other approaches to classify text concern the use of external information to add metadata to each of the tokens. This is the case of [11] that uses data repositories like Wikipedia to improve the performance of clustering algorithms. However, this approach is time consuming when the reality is that the analysis of social media has to be done in "real time".

In [10] a difference is made between the concepts of author attribution and author profiling. The first one concerns the identification of authorship of an anonymous piece of writing while the second one is attributed to the study of certain linguistic features that vary according to the author. So, in [10] the authors are classified according to their characteristics.

After reviewing some recent studies in the area of text classification and authorship attribution, at this state of our study, no works are found focused on the identification of the author within a corpus that *a priori* comes from a single author and it is already tagged.

In this paper, an authorship recognition method in tweets by using Naïve Bayes algorithm is presented. The proposed method is validated by using a confusion matrix in 4 political public profiles mixed with more users. The next section describes each of the four steps of the process starting with the retrieval of tweets to compose a corpus.

3. Experimental approach

A framework for experimentation, based on Naïve Bayes, was developed in order to meet the goal outlined in the introduction. With Naïve Bayes, any vector that represents a tweet will have to contain information about the probabilities of appearance of the words of the text within the tweets, of a given public profile, so that the algorithm can compute the likelihood of that tweet belonging to the author.

Our approach consists of the following four steps: 1) Data import, 2) Data processing, 3) Prediction model and 4) Evaluate Results. For all the process, R language (https://www.r-project.org/) was used because of the facilities given by the some of their packages and functions.

3.1. Data import

The data import concerns the recovery of the tweets by specifying a user profile. R, and its library *twitteR*, has been used in order to connect to the Twitter platform. By specifying keys provided by the Twitter developer page it is possible to access the Application Program Interface (API).

The main structure for managing tweets is by creating a corpus which is a set of tweets that contains similar characteristics. In this case, all the tweets that compose the corpus were retrieved the same day as 4 of the accounts belong to politicians.

One of the previous steps before storing the tweets is the preprocessing step which allows the elimination of data that is not of interest. In this case, all the URL were removed because they have been shortened and so they don't provide any significance.

3.2. Data processing

To classify text using Naïve Bayes, the data need to be storage in a data frame which is used for storing data tables. It is a list of vectors of equal length. Also, this data must have a specific structure:

- each row must correspond to a specific tweet,
- each column must correspond to a specific word,
- each cell must indicate if a word appears in a specific tweet.

All the data, that is storage in the data frame, has to be converted into a sparse matrix so it is necessary to:

- segment each tweet by words,
- create one new column for each word,
- count how many times each word appears in each tweet,
- translate the matrix from a "high" format to a "wide" format, this means to have as many columns as words.

R provides some functions that help to convert the data frame (see Fig. 1) into a sparse matrix (see Fig. 2). These functions are *tidytext*, *dplyr* and *tidyr*.

```
# A tibble: 9,394 x 3
   id                                screenName                text
   <chr>                             <chr>                     <chr>
 1 x2897441                          EPN                       #MiBandera  @TerminalNavega https://t.co/…
 2 x166669818                        VicenteFoxQue             "\"Avanzando hacia una crisis...."
 3 "Vemos las gigantescas perdidas de… ""                       ""
 4 Hacia mucho tiempo no pasaba esto!! ""                       ""
 5 "Muchos frentes equivocados: "      ""                       ""
 6 Queremos que acabe con la corrupci… " sin empleo y sin Pa<U+00E… ""
 7 x82119937                         lopezobrador_             <U+00A1>Miren cu<U+00E1>nta gente partici…
 8 x166669818                        VicenteFoxQue             Situaci<U+00F3>n que a<U+00FA>n no ocurre…
 9 x82119937                         lopezobrador_             "\"Inauguraci<U+00F3>n del estadio de b<U…
10 " desde Ciudad de M<U+00E9>xico. h… ""                       ""
# … with 9,384 more rows
```

Fig. 1. Initial data frame composed by 9394 rows and 3 columns (*id*, *screenName* and *text*).

```
# A tibble: 863 x 11,011
   screenName id    `_deepakanand` `_iga` `_paolimeow` `_siqure`  `_v`   `0` `0.2` `0.6`  `00` `000`
   <chr>      <chr>          <int>  <int>        <int>     <int> <int> <int> <int> <int> <int> <int>
 1 "  @lopez… " cu…             NA     NA           NA        NA    NA    NA    NA    NA    NA    NA
 2 "  arriba… "  l…             NA     NA           NA        NA    NA    NA    NA    NA    NA    NA
 3 "  deja d… L<U+…             NA     NA           NA        NA    NA    NA    NA    NA    NA    NA
 4 "  DESHON… CORR…             NA     NA           NA        NA    NA    NA    NA    NA    NA    NA
 5 "  especi… " an…             NA     NA           NA        NA    NA    NA    NA    NA    NA    NA
 6 "  no es … Gobe…             NA     NA           NA        NA    NA    NA    NA    NA    NA    NA
 7 "  peores… " la…             NA     NA           NA        NA    NA    NA    NA    NA    NA    NA
 8 "  podr<U… Si e…             NA     NA           NA        NA    NA    NA    NA    NA    NA    NA
 9 "  todos … Enti…             NA     NA           NA        NA    NA    NA    NA    NA    NA    NA
10 " @CeciTe… " @C…             NA     NA           NA        NA    NA    NA    NA    NA    NA    NA
# … with 853 more rows, and 10,999 more variables: `0001f1e6` <int>, `0001f1e7` <int>, `0001f1e8` <int>,
#   `0001f1ea` <int>, `0001f1f1` <int>, `0001f1f2` <int>, `0001f1f7` <int>, `0001f1f8` <int>,
#   `0001f1fa` <int>, `0001f1fb` <int>, `0001f1fd` <int>, `0001f399` <int>, `0001f3ad` <int>,
#   `0001f3eb` <int>, `0001f3fb` <int>, `0001f3fc` <int>, `0001f3fd` <int>, `0001f41f` <int>,
#   `0001f444` <int>, `0001f447` <int>, `0001f449` <int>, `0001f44b` <int>, `0001f44c` <int>,
#   `0001f44d` <int>, `0001f44f` <int>, `0001f468` <int>, `0001f469` <int>, `0001f49a` <int>,
#   `0001f4a1` <int>, `0001f4a5` <int>, `0001f4ba` <int>, `0001f4c9` <int>, `0001f4e1` <int>,
```

Fig. 2. Sparse matrix converted from the initial data frame, with 863 rows (one for each tweet) and 11011 columns (one for each word).

3.3. Prediction model

In machine learning, there is a big interest in selecting the best hypothesis given certain data. So, in a classification problem the hypothesis may be the class to assign for a new data instance. The Bayes' Theorem provides a way to calculate the probability of a hypothesis given prior knowledge. Naïve Bayes can be used for multi-class classification problems, however in this case the classification was started with a binary (two-class).

In the creation of a classification model a diagnosis is needed of how well it is doing its job. So, to achieve this purpose the data have to be divided into two sets, one of training and one of test. With the training set the model will be adjusted, in this case, by determining conditional probabilities of each word, for each category. Then, this model is applied in the test set to be able to analyze how many of the cases were correctly classified.

For the experimental approach, the data was divided, 70% in the training set and the rest in the test one. A random sample of the data is used. A list of probabilities is stored to a file for a learned Naïve Bayes model. This includes:

– class probabilities: the probabilities of each class in the training dataset, which are the frequency of words that belong to each author (class) divided by the total number of instances.

– conditional probabilities: the conditional probabilities of each input value given each class value. These probabilities are the frequency of each attribute value for a given class value (profile author) divided by the frequency of instances (words) with that class value.

When using the Naïve Bayes function, with the train set, it asks for the data that is going to be used. Also, the objective variable is an argument that have to be given in order to proceed to the classification. In this case, see Fig. 1, the objective or dependent variable corresponds to the *screenName*. All the other variables will be the predictors or independent variables. The model allows to make predictions by using the function *predict()* that is contained in the R library named *base*. For this the test set is used.

3.4. Evaluate results

The evaluation of the results is done by using a confusion matrix which is a table that describes the performance of a classification model on a set of test data for which the true values are known.

The *caret* library for machine learning, in R, can calculate a confusion matrix with the function *confusionMatrix()*. This matrix allows to analyze the accuracy of the predictions and some adjustment measures. The metrics shown in the confusion matrix are:

– Accuracy: represents how often is the classifier correct, is the fraction of predictions the model got right.
– 95% CI: confidence interval with a degree of confidence of 95%.
– *P*-Value: one-sided test to see if the accuracy is better than the "no information rate," which is taken to be the largest class percentage in the data.
– Kappa: measure of how much the model improves a prediction against the observed probabilities.
– Mcnemar's Test *P*-Value: captures the errors made by both models (training and test).
– Sensitivity: proportion of positive results out of the number of samples which were actually positive.
– Specificity: proportion of truly negative cases that were classified as negative; it is a measure of how well the classifier identifies negative cases.
– Positive Predictive Value: the percent of predicted positives that are actually positive.
– Negative Predictive Value: the percent of negative positives that are actually negative.
– Prevalence: the ratio of actual yes to total number of instances.
– Detection Rate: the rate of true events also predicted to be events.
– Detection Prevalence: the prevalence of predicted events.
– Balanced Accuracy: the measure of how accurate is the overall performance of a model considering both positive and negative classes without worrying about the imbalance of a data set.

The confusion matrix has to be analyzed in order to interpret it in a correct way. Some metrics could be more useful to understand how the model is working.

In the next section is presented a case of use based on a corpus mainly composed of 4 different public profiles of politicians in Mexico and others users extracted from the hashtag #*UniformeNeutro*.

4. Case of use and results

The case of use presented in this article concerns the analysis of 4 public profiles of politicians in Twitter in order to classify them according to their tweets. The main interest is to discover in what cases a tweet is attributed to the politician according to the words used and in what other cases it is classified as *other* so it is possible to affirm that the tweet could have been written by someone else.

The classification model was implemented in R because of the facilities given through the use of their libraries like: *twitteR*, *tidyverse*, *tidytext*, *naivebayes* or *caret*.

For the case of use, the corpus is composed of tweets from 4 Mexican political public profiles (@*lopezobrador_*, @*EPN*, @*FelipeCalderon* and @*VicenteFoxQue*) as other users not concerning the political field.

To be able to analyze the tweets it was important to follow all the steps explained in Section 3. In the data import step, 4185 tweets were retrieved by specifying each of the profiles and then the hashtag #*UniformeNeutro*, an initiative of the Mexican government to allow girls and boys to choose between pants or skirts to use in the school. Obviously

```
 [1] VicenteFoxQue Other         VicenteFoxQue VicenteFoxQue Other         VicenteFoxQue VicenteFoxQue
 [8] VicenteFoxQue VicenteFoxQue VicenteFoxQue Other         Other         VicenteFoxQue VicenteFoxQue
[15] VicenteFoxQue VicenteFoxQue VicenteFoxQue VicenteFoxQue VicenteFoxQue VicenteFoxQue Other
[22] VicenteFoxQue VicenteFoxQue VicenteFoxQue VicenteFoxQue Other         Other         Other
[29] VicenteFoxQue VicenteFoxQue VicenteFoxQue VicenteFoxQue Other         Other         VicenteFoxQue
[36] VicenteFoxQue VicenteFoxQue VicenteFoxQue VicenteFoxQue VicenteFoxQue VicenteFoxQue VicenteFoxQue
[43] VicenteFoxQue VicenteFoxQue VicenteFoxQue VicenteFoxQue VicenteFoxQue Other         VicenteFoxQue
[50] VicenteFoxQue Other         VicenteFoxQue VicenteFoxQue Other         Other         VicenteFoxQue
[57] Other         VicenteFoxQue VicenteFoxQue VicenteFoxQue VicenteFoxQue VicenteFoxQue Other
[64] Other         VicenteFoxQue
Levels: Other VicenteFoxQue
```

Fig. 3. Vector obtained with the *screenName* values that have been predicted by the model.

due to the controversy, the tweets are variated with messages addressing the government. The data import concerns, also, the replacement of badly downloaded characters so the tweets can be processed without problems. After that, the table that contains the tweets is transformed into a data frame by specifying that only it will be retrieved the columns concerning the *id*, *screen name* and *tweet*. From the data frame, it is possible to create a sparse matrix where each row stands for a tweet and each column for a word (see Figs. 1 and 2). For all the political profiles the number of tweets are exactly the same so each matrix is composed of 675 rows and 9485 variables representing each word. These matrices tend to get very big, so the case of use is only an example in order to be able to manipulate the data and analyze it correctly.

The prediction model aims to estimate when a tweet belongs to a specific user. For this test, the model tries to predict whether a tweet was made by one of the 4 political authors accounts or not. Since the others profiles are not of interest they are tagged with the label "*other*".

As explained in the previous section, to be able to adjust the model a training and a test set is used. Also, the function *naivebayes()* is applied with the training set. With this model, the predictions are made. The prediction model is applied to the 4 accounts. In Fig. 3, there is an example of the vector obtained for the first 65 tweets, when using a prediction for the public profile *@VicenteFoxQue*.

To evaluate the success of the prediction model, the function *confusionmatrix()* receives the vector with the predictions and the actual values of *screenName* (the name of the account). This process was made for the 4 profiles.

To analyze the results obtained, the Figs. 4, 5, 6 and 7 and Tables 1, 2, 3, and 4 will be used. The initial

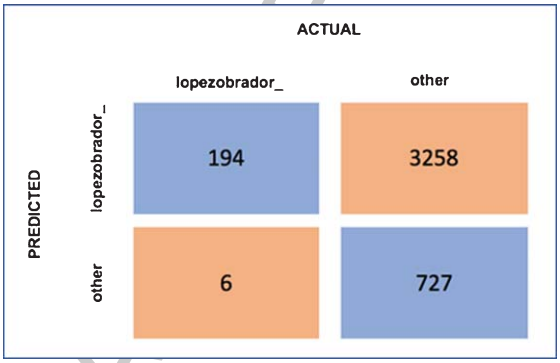

Fig. 4. Visualization of the confusion matrix for the public account *@lopezobrador_*.

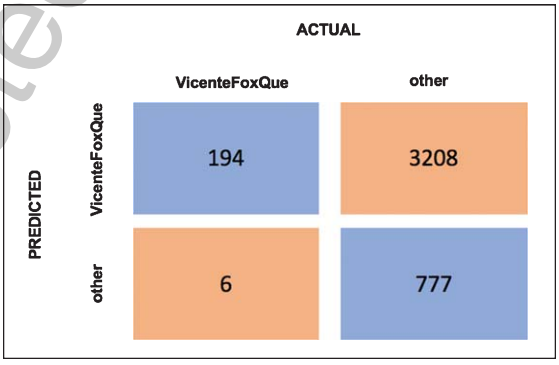

Fig. 5. Visualization of the confusion matrix for the public account *@VicenteFoxQue*.

prediction model was used but it was adjusted after several tests.

4.1. Results for @lopezobrador_

From 200 cases identified for the account *@lopezobrador_*, in the test set, 194 were correctly classified

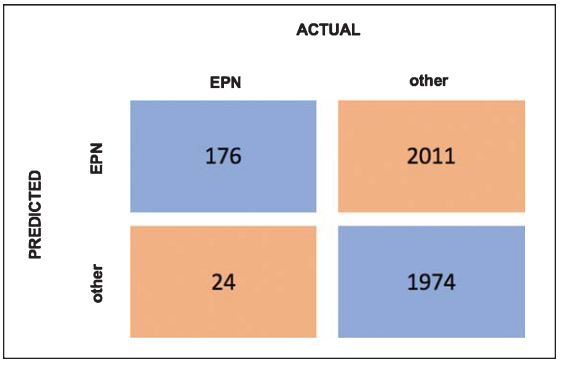

Fig. 6. Visualization of the confusion matrix for the public account *@EPN*.

Table 1
Metrics obtained in each confusion matrix for the public account of *@lopezobrador_*

	@lopezobrador_
Accuracy:	0.7353
95% CI:	(0.7272, 0.7433)
No Information Rate:	0.6989
P-Value [Acc > NIR]:	<2.2e-16
Kappa:	0.1987
Mcnemar's Test *P*-Value:	<2.2e-16
Sensitivity:	0.9735
Specificity:	0.1824
Pos Pred Value:	0.7343
Neg Pred Value:	0.7480
Prevalence:	0.6989
Detection Rate:	0.6804
Detection Prevalence:	0.9266
Balanced Accuracy:	0.5780
'Positive' Class:	@lopezobrador_

(Fig. 4). This means, a sensitivity of 97% (Table 1), only 6 were classified as *other*. For the accounts classified as *other*, of a total of 3985, only 727 were correctly classified, that is, 18% of specificity.

Table 1 shows the metrics obtained for *@lopezobrador_* with a precision (accuracy) of 73%.

Another useful measure corresponds to the Kappa statistic which gives a measure of how much the model improves a prediction against the observed probabilities. The closer to 1 is the Kappa value it means that the model is better than the expected probability. For this analysis, the values that are considered ideals are situated above 0.6. In this case of use, the prediction model was ameliorated and also the Kappa value, this is shown as the different examples are presented in the next subsections. The Kappa value for the prediction model used for *@lopezobrador_* got a value of 0.1987 which was still very low.

Table 2
Metrics obtained in each confusion matrix for the public account of @VicenteFoxQue

	@VicenteFoxQue
Accuracy:	0.8804
95% CI:	(0.8748, 0.8858)
No Information Rate:	0.8816
P-Value [Acc > NIR]:	0.6705
Kappa:	0.2261
Mcnemar's Test *P*-Value:	<2e-16
Sensitivity:	0.9725
Specificity:	0.1949
Pos Pred Value:	0.8999
Neg Pred Value:	0.4875
Prevalence:	0.8816
Detection Rate:	0.8573
Detection Prevalence:	0.9527
Balanced Accuracy:	0.5837
'Positive' Class:	@VicenteFoxQue

The positive predictive value (Pos Pred Value), as explained in the previous section, indicates the probability that a data that has been predicted as belonging to the positive category, really belongs to it ('Positive' class: *@lopezobrador_*, in this example). In this case, *@lopezobrador_*, the probability is of 73.43%. By complement, the negative predictive value (Neg Pred Value) indicates the probability that a data predicted as belonging to the negative category (*other*), in effects corresponds to that. The negative predictive value obtained was of 74.8%.

Finally, the balanced precision indicates how well the model predicts both positive and negative categories. This is very important with data, like the presented here, where the classes are unbalanced, that is, one is more abundant and more likely to appear than the other. In data sets like these, it is easy to obtain a high accuracy (73%) for the most probable class, even if there is low for the less probable class. The balanced accuracy is 57.80% which have to be improved.

4.2. Results for @VicenteFoxQue

Similar to the results of *@lopezobrador_*, with *@VicenteFoxQue* 194 were correctly classified (Fig. 5) obtaining a sensitivity of 97% (Table 2), only 6 were classified as *other*. For the accounts classified as *other*, 777 were correctly classified, which means that 19% corresponds to specificity.

The Kappa value for the prediction model used for *@VicenteFoxQue* is better than the one obtained in *@lopezobrador_* test, with a value of 0.2261. However, this value was still low.

Table 3
Metrics obtained in each confusion matrix for the public account of @EPN

	@EPN_
Accuracy:	0.775
95% CI:	(0.7309, 0.815)
No Information Rate:	0.7225
P-Value [Acc > NIR]:	0.00994
Kappa:	0.4024
Mcnemar's Test *P*-Value:	0.02686
Sensitivity:	0.8824
Specificity:	0.4955
Pos Pred Value:	0.8199
Neg Pred Value:	0.6180
Prevalence:	0.7225
Detection Rate:	0.6375
Detection Prevalence:	0.7775
Balanced Accuracy:	0.6889
'Positive' Class:	Other

Table 4
Metrics obtained in each confusion matrix for the public account of @FelipeCalderon

	@FelipeCalderon
Accuracy:	0.9787
95% CI:	(0.9704, 0.9851)
No Information Rate:	0.9561
P-Value [Acc > NIR]:	6.456e-07
Kappa:	0.6746
Mcnemar's Test *P*-Value:	6.33e-08
Sensitivity:	0.9994
Specificity:	0.5278
Pos Pred Value:	0.9788
Neg Pred Value:	0.9744
Prevalence:	0.9561
Detection Rate:	0.9555
Detection Prevalence:	0.9762
Balanced Accuracy:	0.7636
'Positive' Class:	@FelipeCalderon

The positive predictive value and the negative predictive value are 89% ('Positive' class: *@VicenteFoxQue*) and 48% respectively.

The accuracy obtained is of 88% while the balanced accuracy is 58.37% which have to be improved.

4.3. Results for @EPN

The results for *@EPN* are kindly different from the ones obtained with *@lopezobrador_* and *@VicenteFoxQue*. In this case, 176 were correctly classified (Fig. 6) with a sensitivity of 88% (Table 3), having 24 classified as *other*. For *other*, 1974 were correctly classified, which means that the specificity is almost of 50%.

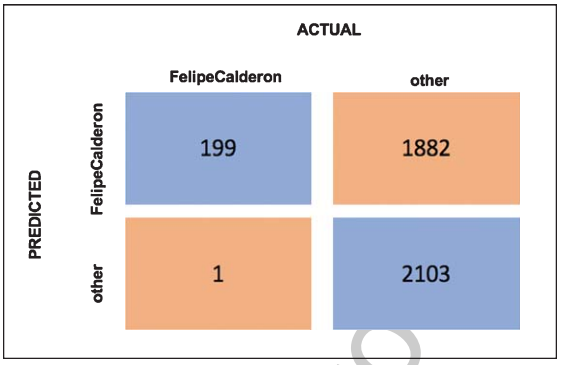

Fig. 7. Visualization of the confusion matrix for the public account *@FelipeCalderon*.

Unlike the two Kappa values presented above, the obtained for *@EPN* shows a considerable improvement with a value of 0.4024 (see Table 3).

The positive predictive value and the negative predictive value are 81% and 61% respectively. However, in this example, the 'Positive' class corresponds to *other*.

The accuracy obtained is of 77% while the balanced accuracy is 68.89% which is bigger than the two previous examples.

4.4. Results for @FelipeCalderon

After some training and adjustment of the prediction model, it shows a better performance comparing it with the results obtained in the 3 previous cases. So, for *@FelipeCalderon* 199 were correctly classified (Fig. 7) with a sensitivity of 99% (Table 4) and only one classified as *other*. For *other*, 2103 were correctly classified, which means that the specificity is of 52%.

In both cases of the predictive value, the positive and the negative, the values are close to the 97%. The 'Positive' class corresponds to *@FelipeCalderon*.

The accuracy obtained is of 97% while the balanced accuracy is 76.36% which is a very good value in the last case of use by using a model improved thanks to the above results.

Finally, the prediction model can be considered as acceptable, or ideal, the Kappa value is of 0.6746 (see Table 4).

5. Conclusions and further work

Text classification on Twitter recently attracted research interest in politics using Information Retrieval and NLP. However, the vast majority of work related to the automatic classification of text has been focused on long texts. Twitter imposes new challenges in working with unstructured data. The objectives of text classification are very different, some of them tend to classify text into classes or categories in order to identify main themes.

In this article, text classification is based on the analysis of tweets because, unlike other information sources, Twitter is up-to-date and reflects the news and events occurring daily all over the world. The interest of analyzing tweets concerns the authorship identification regardless that, in politician accounts, the author is already known. However, its well known that these accounts are manage by a group of people and not always is the same person who creates the tweet. So, the present work is interested in finding some clues to identify authorship and the number of people working behind the scenes after applying a classification model based on Naïve Bayes' theorem.

In this article, a framework for experimentation based on Naïve Bayes was implemented by using R. Given a public profile, of a politician, the tweets are retrieved, processed and classified according to the author or *other*. The analysis and interpretation of the results are possible thanks to a confusion matrix, with different metrics, created for each profile. Experimental results show that with only a small set of features, the classifier achieves a significant improvement in accuracy. However, there is an opportunity to continue the work in testing the classifier accuracy and performance at larger scales, as well as in several other areas. While the results with four authors are promising, a Kappa value of 67%, accuracy of 97% and balanced accuracy of 76%, real-world applications require thousands of authors. Also, it would be interesting to determine when accuracy falls when using a bigger corpus.

Future work should consider the use of other classifiers apart of Naïve Bayes. Also, a comparison with results obtained in other authorship detection research is desirable. This comparison must include results like the ones obtained in the international competition PAN (http://pan.webis.de).

References

[1] A. Castro and B. Lindauer, Author Identification on Twitter, 2012.

[2] B. Sriram, Short text classification in twitter to improve information filtering. Doctoral dissertation, The Ohio State University. 2010.

[3] D. Murthy, Twitter: Social Communication in the Twitter Age. Cambridge, UK: Polity Press, (2013), pp. 193.

[4] E. AlSukhni and Q. Alequr, Investigating the use of machine learning algorithms in detecting gender of the Arabic tweet author, *International Journal of Advanced Computer Science & Applications* **1**(7) (2016), 319–328.

[5] J.D. Burger, J. Henderson, G. Kim and G. Zarrella, Discriminating Gender on Twitter. Proceedings of the Conference on Empirical Methods in Natural Language Processing. EMNLP '11, Association for Computational Linguistics, Stroudsburg, PA, USA. (2011), pp. 1301–1309.

[6] M. Sultana, P. Polash and M. Gavrilova, Authorship recognition of tweets: A comparison between social behavior and linguistic profiles, *IEEE International Conference on Systems, Man, and Cybernetics (SMC)* (2017), 471–476.

[7] P. Galán-García, J.G.D.L. Puerta, C.L. Gómez, I. Santos and P.G. Bringas, Supervised machine learning for the detection of troll profiles in twitter social network: Application to a real case of cyberbullying, *Logic Journal of the IGPL* **24**(1) (2016), 42–53.

[8] R. Green and J. Sheppard, Comparting frequency- and style-based features for twitter author identification, *Proceedings of the International Florida Artificial Intelligence Research Society Conference (FLAIRS)*, 2013.

[9] R. Sousa Silva, G. Laboreiro, L. Sarmento, T. Grant, E. Oliveira and B. Maia, 'twazn me!!!;(' Automatic authorship analysis of micro-blogging messages. Muñoz R., Montoyo A., Métais E. (eds) Natural Language Processing and Information Systems, NLDB 2011. Lecture Notes in Computer Science, vol 6716. Springer, Berlin, Hidelberg. (2011), pp. 161–168.

[10] S. Mechti, M. Jaoua, L.H. Belguith and R. Faiz, Machine learning for classifying authors of anonymous tweets, blogs, reviews and social media. Proceedings of the PAN@ CLEF, Sheffield, England. 2014.

[11] X.H. Phan, L.M. Nguyen and S. Horiguchi, Learning to classify short and sparse text & web with hidden topics from large-scale data collections, *Proceedings of the World Wide Web*, ACM, Beijing, China, (2008), 91–100.