Stacked Gender Prediction from Tweet Texts and Images

Notebook for PAN at CLEF 2018

Giovanni Ciccone*, Arthur Sultan*,**, Léa Laporte*, Előd Egyed-Zsigmond*, Alaa Alhamzeh*,**, and Michael Granitzer**

* Université de Lyon - INSA Lyon - LIRIS UMR5205, ** Universität Passau
giovanni.ciccone.1994@gmail.com, arthur.sultan@insa-lyon.fr,
elod.egyed-zsigmond@insa-lyon.fr, lea.laporte@insa-lyon.fr, alaa.alhamzeh@insa-lyon.fr,
michael.granitzer@uni-passau.de

Abstract This paper describes our participation at the PAN 2018 Author Profiling shared task. Given texts and images from some Twitter's authors, the goal is to estimate their genders. We considered all the languages (Arabic, English and Spanish) and all the prediction types (only from texts, only from images and combined). The final submitted system is a stacked classifier composed of two main parts. The first one, based on previous PAN Author Profiling editions, concerns gender prediction from texts. It consists in a pipeline of preprocessing, word n-grams from 1 to 2, TF-IDF with sublinear weighting, Linear Support Vector classification and probability calibration. The second part is formed by different layers of classifiers used for gender estimation from images: four base classifiers (object detection, face recognition, colour histograms, local binary patterns) in the first layer, a meta classifier in the second layer and an aggregation classifier as third layer. Finally, the two gender predictions, from texts and images, feed into the last layer classifier that provides the combined gender predictions.

1 Introduction

The prediction of the gender of an author is part of a more general task called author profiling. Author profiling aims at predicting the characteristics of an author (age, gender, social background, etc...) based on content produced by the author. Author profiling is useful for marketing intelligence, to analyze customers characteristics [10] [12] . Author profiling can also be used in forensics, in order to set up a suspect profile from a threat or sexual harassment document [5]. For example, in 2001, Roger Shuy analyzed a ransom note which led to the arrest of its author [9]. Another possible application for author profiling is in the field of security, for example to detect emails written by terrorists, from an established standard profile [7].

Author profiling from tweets has been studied since at least 2013, through research tasks proposed by the PAN annual challenge [11]. However, until now, the prediction was based only on text taken from social medias, while this year, images were added to the available data. The objective of the 2018 Pan author profiling shared task and of our approach is thus to study if and how text and images taken from tweets can be used to predict the gender of the authors of those tweets [13].

To do so, we built two independent classifiers, with for each, as output, the probability of an author to belong to the "male" or to the "female" class. The first classifier uses textual features whereas the second classifier uses image-based features. Finally, a meta-classifier combines the prediction of the text-based classifier and of the image-based classifier in order to provide a prediction based on the combination of the textual and image-based features.

This paper is structured as follows. In section 2, we present consecutively the functioning of our text-based, image-based and meta classifiers. Section 3 deals with the results of our approach, obtained on the PAN 2018 author profiling test dataset. Finally, we draw the conclusion of our work in section 4.

2 Overview of Our Proposed Method

2.1 Gender Prediction from Tweet Texts

The requirement of predicting users' gender from textual Twitter data was proposed also in PAN 2017 edition of Author Profiling shared task so we based our work on the analyses performed by previous edition participants [14]. The idea is to grab hints from their works in order to reach similar results as quick as possible and devote the remaining part of the available time for handling the novelty of this year's challenge that concerns gender prediction from images. Regarding the text sub-task, we mainly based our work on the papers written by two teams of PAN 2017 Author Profiling task: the winner [3] and our research team LIRIS [8].

The approach used by us in PAN 2018 challenge for gender prediction from tweets texts consists in a pipeline formed by: text preprocessing, n-gram Bag of Words (BoW), Term Frequency-Inverse Document Frequency (TF-IDF) weighting, Linear Support Vector Classification (LinearSVC) and probability calibration (CalibratedClassifierCV). Figure 1 shows the proposed method. In the following we present in detail the different steps.

Preprocessing 's goal is to filter useless data in order to obtain a smaller dataset and consequently reduce resource usage and computation time. Firstly, we analyzed the provided data for having some hints regarding which preprocessing actions to implement and how. Table 1 describes counts of interesting preprocessing characteristics regarding the training dataset.

The provided dataset has two kinds of difficulties:

- the first one is the presence of three very different languages (Arabic, English and Spanish). They differ for alphabet, syntax and grammar rules.
- the second issue is related to tweets nature, the shortness (maximum 140 characters), they often contain hashtags, user mentions, URLs, slang expressions, misspelled words, poor grammar.

We did some prototypes by using Python nltk tweets tokenizer, called *TweetTokenizer*. For Arabic language, we discovered that it has some difficulties in handling diacritics, that are a kind of accents used in Arabic language. Specifically, the tokenizer,

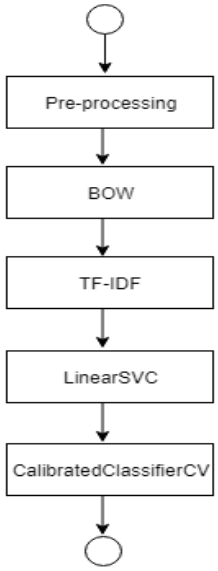

Figure 1. Proposed method for gender profiling from tweets texts

Name	Language	Number
URLs	Arabic	44556
URLs	English	138128
URLs	Spanish	118957
users' mentions	Arabic	66917
users' mentions	English	238599
users' mentions	Spanish	217703
punctuation signs	Arabic	1039379
punctuation signs	English	1826354
punctuation signs	Spanish	1549798
stopwords	Arabic	237753
stopwords	English	1254179
stopwords	Spanish	1367597
diacritics	Arabic	120742
diacritics	English	—
diacritics	Spanish	—

Table 1. Preprocessing characteristics counts

when finding a diacritic, splits the word in three tokens: the part before the diacritic, the diacritic itself and the part after. This behaviour leads to worse results therefore we implemented a script for Arabic text normalization and tokenization by taking into account this issue. For what concern preprocessing tweets features we did some considerations about URLs ('http://...') and user mentions (@user). The point is that they don't carry on information that can be used for inferring the author's gender, therefore we decided to filter them.

Several techniques have been proposed in literature for tweets preprocessing before further information extraction [16]. Considering those approaches and our analyses on the training dataset we decided to use the preprocessing architecture shown in figure 2. Independently from the language, we apply the HTML unescaping and filtering of URLs and user mentions. Since they are not correlated to author's gender and language, this operation can be done at the beginning independently from the language. Afterwards, for English and Spanish texts, the following actions are performed: removal of punctuation, repeating characters and stopwords. These operations are applied also to Arabic corpus in addition to textual normalization and diacritics removal.

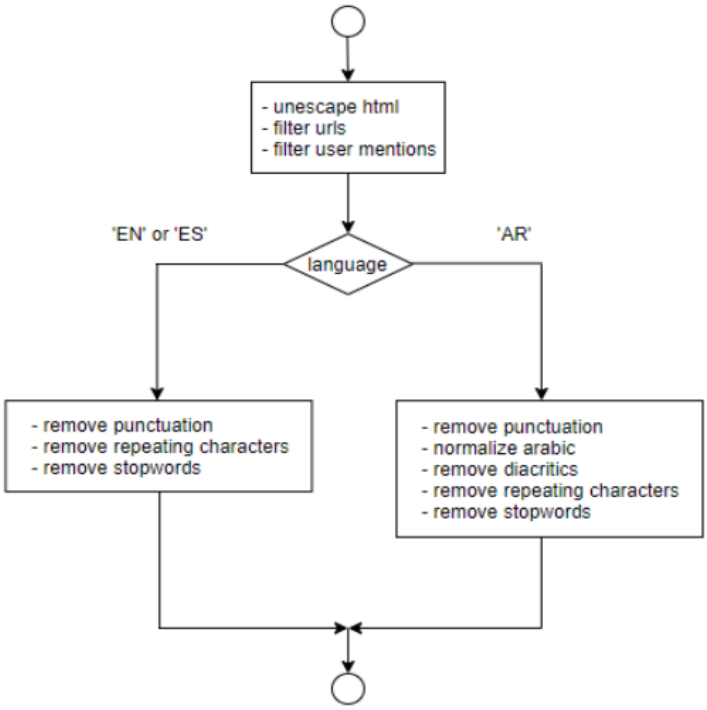

Figure 2. Text preprocessing proposed architecture

Feature extraction: This stage converts text data to vectors of floats representing the scores of tokens within each document. One document contains all the 100 tweets for a certain author. We considered the representation of tweets based on n-grams and TF-IDF. N-gram BOW model consists in representing a text as a multi-set (bag) of its tokens (for instance words) ignoring the grammar and the order of the words but taking in account only the multiplicity. TF-IDF is a well-known technique used in Information Retrieval that produces scores depending on the number of token occurrences within a document and on the number of distinct documents containing the tokens. TF-IDF simplifies learning algorithms in selecting more discriminative words. This technique has been widely used for the Author Profiling task. Our implementation relies on *CountVectorizer* and *TfidfTransformer* libraries of Python sklearn. For BoW we tested 2 methods:

1. the approach used by PAN 2017 winner that consists in a combination of character n-grams from 3 to 5 and word n-grams from 1 to 2
2. word n-grams from 1 to 2

According to our experiments, approach 1 gave more or less the same accuracy results as approach 2 but having the disadvantage of bigger data structures because of the great number of character based n-grams. *CountVectorizer* parameters *analyzer* and *ngram_range* are used for specifying the level of tokens (word level) and the n-grams range (from 1 to 2, that correspond to uni-grams and bi-grams), $min\_df = 2$ means that all the tokens appearing only once are not considered and this causes a matrix dimensions reduction. The *tokenizer* is a reference to the Python method used for tokenizing the text, it can be either the nltk default one or a user defined method. In our case we defined one tokenizer per language reflecting the requirements specified in the figure 2. Concerning *TfidfTransformer*, sublinear_tf weighting conducted to an overall quality improving for each language corpus (around 2% in terms of average macro f score). Sublinear term frequency scaling means to replace tf with $1 + log(tf)$ in TF-IDF formula.

Machine learning algorithms Regarding the classifier, we tried the most commonly ones in PAN past editions, that are Support Vector Machine, Random Forrest, Naive Bayes. As asserted by past teams, the best one for gender classification is Support Vector Machine. More precisely, the linear approach (*LinearSVC*) allows to reach better outcomes than the kernel approach, therefore we decided to use LinearSVC. However, it has the disadvantage of providing only the output labels without the associated probabilities, but in PAN 2018 scenario in which there are two different sub-tasks about texts and images it would be better to have the intermediate outputs (from texts and images) with the corresponding probabilities, this makes easier their combination for obtaining the final results. For this goal of probability calibration we used a *CalibratedClassifierCV* in cascade to *LinearSVC*.

2.2 Gender Prediction from Tweet Images

Our classifier based on images is composed of 3 layers of classifiers. The first layer is composed of classifiers which we will call "low classifiers". Each low classifier is

based on one only type of image feature and outputs the probability that the input image was posted by a male or a female. The second layer is composed of a meta-classifier which combines the prediction of the low classifiers, in order to provide an improved prediction, based on the classifier stacking principle [6]. Finally, the last layer is another meta-classifier which combines the predictions from the second layer, given from the 10 images associated to the author we try to predict the gender.

a) Low Classifiers (Layer 1)

The first layer is composed of 4 independent classifiers. Each of them performs the prediction of the gender of the author of the input image, based on specific image representations. The 4 kind of methods we used are listed below.

- **Object recognition**: Images are represented based on the objects they contain, detected using an object recognition algorithm. The object recognition task is performed with the library YOLO [4], with a confidence threshold of 0.2. We think that an increase the confidence threshold should give better prediction results, but we did not have enough time to study this phenomenon. The feature vector V_{object} resulting from the object recognition task is such as:

$$V_{object} = \{O_1 : I_1, O_2 : I_2, ..., O_i : I_i\},$$

  with O_i an object identified in the image and I_i the "importance" associated to that label. The importance I of an object O is defined such as:

$$I = \sum_{z=0}^{n} confidence(O_z),$$

  with confidence(O_z), the confidence associated to the z^{th} recognition of the object O by YOLO.
  For example, for an image containing two cats and one person, the output of the object recognition task done by YOLO could be: [{ label: "cat", confidence: 0.8 }, { label: "cat", confidence: 0.6 }, { label: "person", confidence: 0.9 }]. The V_{object} resulting feature vector would then be: V_{object} = { "cat": 1.4, "person": 0.9 }.
  One important note, is that we did not spend a lot of time trying to find a good model for the computation of the importance I of an object O. This computation could hence certainly be improved in order to achieve better prediction results.

- **Facial recognition**: Images are represented by two features, respectively the number of men and women detected in the image. We used a neural network which was pre-trained to detect the gender of people in an image, based on their faces only [17]. We thus count the number of male and female faces identified in the input image. The resulting V_{face} vector is such as:

$$V_{face} = \{Male : x, Female : y\},$$

  with x and y respectively the number of male and female faces identified in the image by the pre-trained network.

From a rough hand-made evaluation made on 500 images from the training dataset, in the recognition of male and female faces in the image, the pre-trained network performs a global accuracy of 96% , a recall of 50% for male faces and of 50% for female faces.

- **Color histogram**: Images are represented by a standard color histogram. The resulting V_{color} vector is the 'flattened' version of the color histogram (i.e dim(V_{color} =768)).

- **Local binary patterns**: Computation of a standard vector of local binary patterns, for 24 points and a radius of 8. To compute this vector, we used the skimage library [1]. The result is a vector V_{LBP} such as dim(V_{LBP})=26.

Each low classifier was trained on 56% of the training dataset (42000 images). The images from the 3 language (arabic, english, spanish), where grouped together for the training: in the 42000 images, there were 8400 images from the arabic folder, 16800 from the english folder and 16800 from the spanish folder.
We evaluated the performance of each of the 4 classifiers on the 56% of the training data mentioned above, thanks to a 20-fold cross-validation process. The results are shown in the following array (the classifiers in the "Classifier type" column are those from the sklearn [2] library):

Table 2. Low classifiers prediction results (on training data)

	Mean accuracy	Standard deviation	Classifier type
Object detection	53.3%	1.3%	Linear SVC with default parameters
Face recognition	56.9%	1.2%	Linear SVC with default parameters
Color histogram	51.6%	1.6%	MultinomialNB with default parameters
LBP	53.1%	1.3%	Linear SVC with default parameters

b) Meta-Classifier (Layer 2)

The second layer is composed of one only classifier called "meta-classifier". This meta-classifier takes as input the outputs of the low classifiers of the layer 1, i.e for each low classifier, the probability estimated by this low classifier that the analyzed image was posted by a male or a female. The meta-classifier thus aggregates the results of the first layer in order to provide an improved prediction of the gender of the author of the analyzed image, based on the idea of classifier stacking [6].

This meta-classifier was trained on 16% of the training dataset (12000 images). The images from the 3 language (arabic, english, spanish), where grouped together for the training: in the 12000 images, there was 2400 images from the arabic folder, 4800 from

the english folder and 4800 from the spanish folder.
We evaluated the performance of the meta-classifier on the 16% of the training data mentioned above, thanks to a 20-fold cross-validation process. The results are a mean accuracy of 58.4%, a standard deviation of 2.2%, for a LinearSVC classifier with default parameters.

c) Aggregation Classifier (Layer 3)

The third layer is composed of one only classifier called the "aggregation classifier". As a reminder, for each author of the training or the evaluation dataset, 10 images are associated to this author. The aggregation classifier takes as input the 10 probabilities that the author is a male or a female, given by the second layer on the 10 images associated to the author. The aim of the aggregation classifier is thus to predict the gender of the author, based on the whole set of genders predicted from the analysis of the 10 images associated to this author.

The aggregation classifier was trained on 8% of the training dataset (600 images). The images from the 3 language (arabic, english, spanish), where grouped together for the training: in the 600 images, there was 120 images from the arabic folder, 240 from the english folder and 240 from the spanish folder.
We evaluated the performance of the aggregation classifier on the 8% of the training data mentioned above, thanks to a 20-fold cross-validation process. The results are a mean accuracy of 69.8%, a standard deviation of 7.7%, for a MultinomialNB classifier with default parameters.

2.3 Gender Prediction from both Texts and Images

Gender prediction from both text and images is done by a classifier we call the "final classifier". This classifier takes as input the outputs of the text and image classifiers (for images, the output of the third layer is used). The aim of the final classifier is hence to combine the gender prediction of the author based on the text associated to the author, and the gender prediction based on the image associated to the author, in order to output a final improved prediction, based on the classifier stacking idea [6]. The two inputs of this final classifier coming from the text and image classifiers are both probabilities.

The final classifier was trained on 20% of the training dataset (1500 authors): 300 arabic authors, 600 english authors and 600 spanish authors. The machine learning algorithm of the final classifier is LinearSVC with default parameters.
We evaluated the performance of the final classifier on the 20% of the training data mentioned above, thanks to a 20-fold cross-validation process. We performed the same evaluation for the text classifier and the image classifier (i.e the "aggregation classifier"), but respectively on 80% and 8% of the training data. The results are shown in the following array:

From this table, we can say that our initial estimations of our classifiers performance, based on the training data, does not allow to conclude that our combined approach improved the prediction score. Indeed, the performance of the final classifier did not give a better prediction result than the prediction of the text classifier.

	Mean accuracy	Standard deviation
Text classifier	80.5%	3.9%
Image classifier	69.8%	7.7%
Final classifier	80.1%	4.9%

Table 3. Text, Image and final classifiers prediction results (on training data)

3 Results on the Evaluation Dataset

The official results we obtained are shown in table 4. As you can see, we got about 80% precision for gender prediction only from texts and approximately 70% for image based estimation. Concerning the combined approach we obtained scores slightly better than the ones about texts.

Language	Accuracy (only text)	Accuracy (only images)	Accuracy (combined)
Arabic	0.7910	0.7010	0.7940
English	0.8074	0.6963	0.8132
Spanish	0.7959	0.6805	0.8000

Table 4. Official results for the PAN'18 Author Profiling task

Text related results are very close to the state of the art, represented by [3]. For images, we cannot compare with previous PAN editions because this is the first time in which images requirement is proposed. Those results stick are consistent with the evaluation of our classifiers on the training data, shown in the table 3.

4 Conclusion

Concerning the gender prediction based on text, the proposed method is a pipeline composed of text preprocessing, n-gram BoW, TF-IDF, Linear Support Vector Classification and probability calibration. Our implementation relies on results presented in [14]. Our goal was to reach the state of the art level as soon as possible in order to dedicate the remaining part of the time for approaching the novelty of this edition, that is the gender prediction based on images. Specifically, we based our text related work on papers [3,8], we focused mainly on the preprocessing step by implementing different tokenizers depending on the languages, one of the most noteworthy points is the diacritics handling for Arabic language. Our final score is around 80% and it is pretty near to the best result of previous PAN edition (82.53% by Basile et al [3]).

Regarding the gender prediction based on images only, we can conclude that our overall approach provides significant results, with an accuracy around 70%. Among the image-based features used, the most effective seems to be the "face recognition" feature. Regarding the "meta-classifier", we cannot conclude that stacking low classifiers were

more efficient, since our meta-classifier only improves the prediction score of the classifier based on the face recognition features by 1.5%, with a standard deviation of 2.2% given by the cross-validation process. However, our approach of combining the 10 images associated to an author seems to be efficient, with an improvement of the accuracy around 11% compared to the accuracy given by a classifier (here the "meta-classifier") based on a single image.

To improve the prediction based on images only, one could add new image-based features, such as character recognition. Indeed, some images are photos or screenshots of text (for example a screenshot of a tweet). Another possibility would be to train one classifier for each language. Indeed, as mentioned in section 2.2, we grouped images from all languages during the training, but it is possible that cultural specificities exist among images of one language, which might be useful to predict the gender of an author of this language. Another possibility to improve the prediction based on images would be to improve the object detection process by using pre-trained network trained to detect more object classes (YOLO [4] can only detect 80 object classes).

Regarding the gender prediction based on the combination of text and images, we can conclude that our approach inspired of classifier stacking [6] does not seem to be efficient. Indeed, only a slight improvement of the prediction with the combined approach can be noted, compared to the text approach only, with an average increase of 0.43%, which is not significant enough but matches with the improvement of 0.48% obtained by Sakaki et al. in a similar work [15].

References

1. Skimage. http://scikit-image.org/
2. Sklearn. http://scikit-learn.org/stable/
3. Basile, A., Dwyer, G., Medvedeva, M., Rawee, J., Haagsma, H., Nissim, M.: N-GRAM: New groningen author-profiling model: Notebook for PAN at CLEF 2017. In: CEUR Workshop Proceedings (2017)
4. Darknet: YOLO. https://pjreddie.com/darknet/yolo/
5. De Vel, O., Anderson, A., Corney, M., Mohay, G.: Mining e-mail content for author identification forensics. ACM Sigmod Record 30(4), 55–64 (2001)
6. Džeroski, S., Ženko, B.: Is combining classifiers with stacking better than selecting the best one? Machine learning 54(3), 255–273 (2004)
7. Fund, Q.N.R.: Arabic Author Profiling for Cyber-Security. https://www.prhlt.upv.es/wp/project/2017/arabic-author-profiling-for-cyber-security (2017), [Online; accessed 25 may 2018]
8. Kheng, G., Laporte, L., Granitzer, M.: INSA Lyon and UNI passau's participation at PAN@CLEF'17: Author Profiling task: Notebook for PAN at CLEF 2017. In: CEUR Workshop Proceedings (2017)
9. McMenamin, G.R.: Forensic linguistics: Advances in forensic stylistics. CRC press (2002)
10. Mishne, G., Glance, N.S., et al.: Predicting movie sales from blogger sentiment. In: AAAI spring symposium: computational approaches to analyzing weblogs. pp. 155–158 (2006)
11. PAN: PAN Author Profiling 2013. https://pan.webis.de/clef13/pan13-web/author-profiling.html (2013)

12. Pham, D.D., Tran, G.B., Pham, S.B.: Author profiling for vietnamese blogs. In: Asian Language Processing, 2009. IALP'09. International Conference on. pp. 190–194. IEEE (2009)
13. 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)
14. Rangel, F., Rosso, P., Potthast, M., Stein, B.: Overview of the 5th author profiling task at PAN 2017: Gender and language variety identification in Twitter. In: CEUR Workshop Proceedings (2017)
15. Sakaki, S., Miura, Y., Ma, X., Hattori, K., Ohkuma, T.: Twitter user gender inference using combined analysis of text and image processing. In: Proceedings of the Third Workshop on Vision and Language. pp. 54–61 (2014)
16. Stammatatos, E., Daelemans, W., Verhoeven, B., Juola, P., López-López, A., Potthast, M., Stein, B.: Overview of the 3rd Author Profiling Task at PAN 2015. CLEF 2015 Labs and Workshops, Notebook Papers. CEUR Workshop Proceedings 1391(31), 898–927 (2015)
17. Won, D.: face-classification. https://github.com/wondonghyeon/face-classification