Gender Identification through Multi-modal Tweet Analysis using MicroTC and Bag of Visual Words
Notebook for PAN at CLEF 2018

Eric S. Tellez[1], Sabino Miranda-Jiménez[1], Daniela Moctezuma[2], Mario Graff[1],
Vladimir Salgado[1], and José Ortiz-Bejar[1]

[1] CONACyT-INFOTEC Centro de Investigación e Innovación en Tecnologías de la Información y Comunicación, México
{eric.tellez,sabino.miranda,mario.graff,vladimir.salgado}@infotec.mx,
jortiz@umich.mx

[2] CONACyT-CentroGEO Centro de Investigación en Ciencias de Información Geoespacial A.C., México
dmoctezuma@centrogeo.edu.mx

Abstract This manuscript describes our solution to solve the Author Profiling task at PAN'18. In this edition, the task asks for identifying the user's gender using both their Tweets containing texts and images. We used our MicroTC (μTC) text classification framework to cope with the text problem, and a novel approach to Bag of Visual Words to solve the image classification, designed for this task to solve the image classification. Finally, we tried to improve the final prediction using a combination of both approaches.

1 Introduction

The author profiling problem predicts specific characteristics of an author through the analysis of his/her documents. These traits can be gender, age, personality, native language, among others [13,10]. In particular, PAN@CLEF provides a common platform and common datasets for evaluating Author Profiling systems using text written by social network users, see [13,11]. In the current edition, the gender task consists on datasets in three different languages: Arabic, English, and Spanish; also, the purpose is to tackle a multi-modal problem, that is, it consists of both text and images, as posted by Twitter users [12].

Our contribution solves the text and image profiling independently, in a multilingual perspective, and then merges the solutions using a convex combination. In particular, we use our text classification framework μTC to profile authors based on their published texts; and for the image problem, we design a variant of Bag of Visual Words (BoVW) using the DAISY [20] feature descriptor and a transformation of these features to text, instead of using plain histograms as it is usual in traditional BoVW approaches.

The rest of the paper is organized as follows. Section 2 presents a brief review of works on gender identification. Section 3 describes our approach to solve the problem and our implemented system. Section 4 details our experimental methodology and lists our results. Finally, conclusions are given in Section 5.

2 Related Work

Author profiling is a core task in the PAN contest since 2013 [13]. In PAN'15 and PAN'16, only age and gender classification tasks were considered [14,15]. PAN'17 introduces the language variety aspect while removes the age identification subtask from the competition [11]. In the current edition, PAN'18 [17], only the gender classification is considered, but now the challenge asks for the methods using a multi-modal data, more precisely, text and image messages of the user.

In this context, several and exciting works have been published in the research community, most of them deal with the problem using only the written text by the authors. Such is the case of [7]; here, the authors created a multilingual author profiling corpus of Facebook's users. The user demographic information (age, gender, native language, native region, qualification, occupation, and personality) with public and private messages of the user. The work proposed in [10] uses feature selection and term weighting schemes. These two approaches are based on a proposed method called Personal Expression Intensity which quantifies the amount of personal information contained by a term, where phrases containing singular first-person pronouns define personal information.

Another approaches for gender classification using images typically employ face's images from the user (see for instance [8], [3], and [1]), but in the PAN competition are considered the images posted by the user, that means, these could be selfies, favorite landscapes, cars, animals, among others.

Recent literature shows the possibility to employ visual user content to learn personal attributes such as gender or age. That is the case of the work presented in [22] where the associated categories are used to acquire the posting behavior and then to predict the user's gender. The Bag of Visual Words model uses SIFT features to capture the image's content. For the experiments, 80 profiles were used. The accuracy of this approach is 0.71, achieved using both posting behavior and image content. Another method to extract user's gender from images shared in social media is presented in [9]. The approach consists of a face detector, with the purpose of identifying male and female faces, and object recognizer with 25 categories, looking for picture semantics. The experiments show that some of these categories are more related to females and other with males, e.g., females shared more dogs pictures and males shared more cats pictures. Their result reached a 75.6% of accuracy with 10K users associated with half a million images shared for them.

Our approach deals with both problems separately and, then, we combine predictions. The textual information is tackled with our MicroTC (μTC) framework, and the profiling based on image content was attempted using a variant of Bag of Visual Words (BoVW) that use DAISY features to feed an especially designed clustering and encoding method. Our approaches will be detailed in the following sections.

3 System Description

As in PAN'17, we tackled the text-based author profiling challenge using our text classifier MicroTC (μTC) [18], since it works regardless of both domain and language particularities.

The core idea behind μTC to solve the text classification task, the problem is formulated as a model selection problem, that is, it selects a competitive configuration from a vast universe of possible ones. Each configuration is composed of a list of text transformations (normalizations and generic transformations), a combination of tokenizers, and a weighting schemes. The following steps describe the μTC configuration space and its implicit work-flow:

i. **Preprocessing functions** We use trivalent and binary parameters. The trivalent values can be set to $\{remove, group, none\}$ which means that the term matching the parameter is removed, grouped in set of predefined classes, or left untouched. In this kind of parameters, μTC contains handlers for hashtags, numbers, urls, users, and emoticons. The binary parameters are boolean, and basically, indicate if the parameter is activated or not. In this parameter set, we support for diacritic removal, character duplication removal, punctuation removal, and case normalization.
ii. **Tokenizers** After all text normalization and transformation, a list of tokens should be extracted. We allow to use n-grams of words ($n = 1, 2, 3$), q-grams of characters ($q = 1, 3, 5, 7, 9$), and skip-grams. For skip-grams we allow to select a few tokenizers like two words with gap one, $(2, 1)$, also we allow to use $(2, 2)$, $(3, 1)$. Instead of selecting one or another tokenizer scheme, we allow to select any combination of the available tokenizers, and perform the union of the final multisets of tokens.
iii. **Weighting schemes.** After we obtained a multiset (bag of tokens) from the tokenizers, we must create a vector space. MicroTC allows to use the raw frequency and the TFIDF schemes to weight the coordinates of the vector. It contains a number of frequency filters that were deactivated for this contribution, see [18] for more details.

To evaluate each configuration, we use a Support Vector Machine (SVM) with a linear kernel to perform the final classification. It is well-known that SVM performs excellently for large dimensional input (which is our case), and the linear kernel also performs well under this conditions. We do not optimize the parameters of the classifier since we are pretty interested in the rest of the process. We use the SVM classifier from *liblinear*, Fan et al. [6].

The model selection is lead by a performance function score that is maximized (solved) by a meta-heuristic. The only assumption is that score slowly varies on similar configurations, such that we can assume some degree of *locally concaveness*, in the sense that a local maximum can be reached using greedy decisions at some given point. Clearly, this is not true in general, and the solver algorithm should be robust enough to get a good approximation even when the assumption is valid only with some degree of certainty. From a practical point of view, a configuration is similar to another if structurally vary in a single parameter. We name the set of all similar configurations of m as its neighborhood. Therefore, the core idea is to start from a set of random configurations, evaluate their neighborhoods and greedily move to the most promising set of configurations. This procedure is repeated until some condition is achieved, like the impossibility of improving the score function, or when a maximum number of iterations is reached. There are several meta-heuristics to solve combinatorial optimization problems. In particular, μTC uses two types of meta-heuristics, *Random Search* [4] and *Hill Climbing* [5,2] algorithms. The former consists in randomly sampling $\mathcal{C}$ and selecting the best configuration among that sample. Given a pivoting configuration, the main idea behind Hill Climbing is to explore the configuration's neighborhood and greedily move to the best neighbor. The process is repeated until no improvement is possible. We improve the whole optimization process applying a Hill Climbing procedure over the best configuration found by a Random Search.

The μTC framework is more detailed in [18], and it is available as open source.[3]

Modeling Users In the text problem, we model each user as an array of its text messages, as we did it in PAN'17 [19] following the weighting scheme of entropy+3. We introduced the *entropy+b* term-weighting in our previous PAN'17 participation, which consists in representing each term by the entropy of the term's empirical distribution over the available classes, using a smoothing parameter b, in this case, $b = 3$. More precisely, defined as follows:

$$\mathsf{entropy}_b(w) = \log |C| - \sum_{c \in C} p_c(w, b) \log \frac{1}{p_c(w, b)},$$

where C is the set of classes, and $p_c(w, b)$ is the probability of term w in class c parametrized with b. More detailed,

$$p_c(w, b) = \frac{\mathsf{freq}_c(w) + b}{b \cdot |C| + \sum_{c \in C} \mathsf{freq}_c(w)}.$$

Here, $\mathsf{freq_c}$ denotes the frequency of the term in the class c.

3.1 Author Profiling through Posted Images Content

In this problem, we model each user as the collection of all its images converted as text. In first place, all images were coverted to grayscale format and resized to 400 × 400 pixels. Later, for the process of image to text transformation three

[3] https://github.com/INGEOTEC/microtc

main steps were followed: i) compute feature descriptors for each image (multiple dense vectors per image), ii) create a codebook using an efficient clustering algorithm, and iii) represent each image as a text using the codebook, and finally, iv) perform text classification over the generated text.

A descriptor algorithm computes many vectors for an image such that each vector describes in some sense a region of the image (sub-image). In our case, we use the DAISY [20] descriptor, which is a fast local descriptor that allows fast dense extraction, we can found it in the literature as part of a typical bag of visual words representations; in particular, we used the implementation found in scikit-image [21]. Furthermore, we explore some parameters modifications to get better results.

For the codebook computation, we use a variant of k-means found in the library SemanticWords.jl[4] using approximate nearest neighbors to compute the centroids. Instead of using a Delone partition like the original algorithm, the used one selects a K nearest centroids to compute the next generation of centroids; this makes the computed clustering more robust to noisy data. The text encoding of the image is then performed using the computed codebook as follows, for each image:

- The DAISY feature descriptors are computed, this computes a grid of dense vectors, with a shape that depends on the step and radius parameter of DAISY and the image geometry.
- The text that represents an image is computed as follows: for each DAISY vector u, visited row-wise, left to right, we compute its k nearest centroids, then u is represented by these centroids, using a unique code for each centroid. The order induced by the distance to u is preserved. This procedure composes a visual word $\hat{u}$ for vector u.
- In addition to visual words, we use a unique code to indicate the division among visual words.

This procedure is computed for all images, such that, the computed text will represent each image. We use a Rocchio-like classifier [16] to classify this representation. A Rocchio classifier represents each class with a single vector which is the centroid (mass-center) of all vectors in that class. The classifier is created using the training set, and test examples are labeled with the label associated with the nearest centroid. More precisely, in this notebook, our centroids are c_{female} and c_{male}. The main difference among a traditional Rocchio is that we used a combination of tokenizers, and we use the sum of vectors instead of the mass-center to create centroids which work well since we compute the nearest neighbor using the cosine similarity.

We tried to use our μTC to perform this final classification, but it exhibited a low performance, we presume this is a consequence of the large alphabet and large vocabulary setup found in our image-to-text encoding. Note that the procedure resembles a typical BoVW, but the differences produce significant

[4] https://github.com/sadit/SemanticWords.jl

improvements in the performance. Among the differences with BoVW is the additional support for failure using k nearest centroids for encoding and the notion of syntax, captured due to our q-gram expansion.

3.2 Combining Text and Image-Based Author Profiling Results

The combined prediction between text and images was computed using a convex combination between the prediction of the text and image problem in a separately way,

$$\mathsf{comb}(u) = \alpha T(u) + (1 - \alpha) I(u)$$

Where α is the weight assigned to prediction of the text-classifier, $T(u)$, and $I(u)$ is the profiling prediction based on the image content.

More detailed, $T(u)$ corresponds to the decision function of the underlying SVM inside μTC, i.e., a number between -1 and 1. For the image problem, we normalized the absolute difference of the angles between examples and centroids, recall we use Rocchio for this subtask, so we have:

$$I(u) = \frac{\angle(u, c_{\text{male}}) - \angle(u, c_{\text{female}}) - \mu_D}{\rho_D},$$

where $D \sim |\angle(u, c_{\text{male}}) - \angle(u, c_{\text{female}})|$ is the random variable representing the absolute differences of the distances between u and centroids, μ_D is the mean of D and ρ_D its standard deviation. Please take into account that the order of c_{male} and c_{female} is important, and it is dependent of the corresponding meaning of the decision function of μTC, i.e., in the given definition, -1 corresponds to female and 1 to male.

4 Experiments and Results

The experiments with the training set were run in an Intel(R) Xeon(R) CPU E5-2680 v4 @ 2.40GHz with 28 threads and 256 GiB of RAM running Ubuntu Linux 16.04. The gold-standard were evaluated in the TIRA platform using a virtual machine with 20GiB of RAM and six cores, necessary to parallelize and reduce the running time of computing image features. We use our μTC using the master branch of https://github.com/INGEOTEC/microTC.[5]

We partitioned the full training dataset into two smaller sets, a new training set containing 70% of the users, and a validation set with the rest 30%. The core idea is then to perform cross-validation on the small training set, so we can optimize parameters, and then select best models in the validation set. Note that our participation includes both text and image tasks, and both follow the same partition scheme. The gold-standard was only accessed through the TIRA evaluation platform.

[5] Available under Apache 2 license

4.1 Results for the Text Classification Subtask

To tackle the text classification we use our μTC text classifier, already proved in PAN'17 in the *author profiling* task. In this sense, we tried the following four different paths:

i. determine the best model for each language,
ii. use the best configuration found in one language and create models all datasets in the different languages,
iii. use the configuration models found in PAN'17 for each language and apply it to the new data, and
iv. we use the best configuration in PAN'17 for some language and create models specific for each new dataset

Table 1 show our best results with the text classification approach using our μTC system. Our best macro-F1 results in Arabic is 0.8377, in English is 0.8266, and in Spanish 0.8143. We selected the Arabic@PAN2017 parameters configuration instead of selecting the optimum for each, but it contains two of the three best performing model. Our final system is in bold, see Table 1.

We use 3-fold cross-validation for the model selection procedure. Once the model selection finished, we use the configuration found to train a μTC machine with the whole (small) training set and measure the performance of that classifier on the validation set. Table 1 shows the performance in the evaluation set; the training set in the specified language was used. The set of parameters was computed in different ways, as explained above.

Surprisingly, in the evaluation set, the fourth option using our configuration of parameters found in PAN'17 for the Arabic language created superior models, in average, than other alternatives and configurations. So, we use it for all our final models. The Arabic@PAN'17 [19] configuration indicates that the entire text should be normalized to lower case, it also commands to delete diacritic symbols, consecutive duplicated characters, while punctuation should be kept. Emoticons and hashtags should be left untouched, while numbers and urls should be grouped into a common token (one per option); also, usernames should be deleted. The resulting text should be tokenized using unigrams, bigrams, and three-grams (word n-grams), and characters q-grams with $q = 1$ and $q = 5$, and also skip-grams $(2, 1)$, i.e., three-grams without the central word. The weighting scheme indicate that low and high filtering should not be applied, and commands to use the entropy weighting with a smoothing factor of 3. It can be seen as weird that lower case is recommended on the Arabic language since the concept of upper and lower case is missing, but it is common to found messages, usernames and hashtags in other languages. Please note that the performance difference between the Arabic configuration and the resulting one of applying the parameter optimization for each dataset is of around 1 or 2 points of accuracy in the validation set, so they may perform quite similar, but it is interesting to verify the power of that configuration.

Table 1. Performance scores of our approaches in the evaluation dataset. All models were trained with the specified dataset, but the configuration could be computed using another dataset.

dataset	**method**	**setup**	**accuracy**	**macro-F1**	**macro-Recall**
Arabic	μTC	Arabic@PAN'18	0.7978	0.7978	0.7982
English	μTC	English@PAN'18	0.8089	0.8089	0.8101
Spanish	μTC	Spanish@PAN'18	0.8144	0.8143	0.8145
Arabic	μTC	Arabic@PAN'18	0.7978	0.7978	0.7978
English	μTC	Arabic@PAN'18	0.7967	0.7966	0.7967
Spanish	μTC	Arabic@PAN'18	0.7756	0.7754	0.7756
Arabic	μTC	English@PAN'18	0.7733	0.7731	0.7730
English	μTC	English@PAN'18	0.8089	0.8089	0.8101
Spanish	μTC	English@PAN'18	0.7856	0.7854	0.7855
Arabic	μTC	Spanish@PAN'18	0.7778	0.7777	0.7779
English	μTC	Spanish@PAN'18	0.8256	0.8255	0.8272
Spanish	μTC	Spanish@PAN'18	0.8144	0.8143	0.8145
Arabic	μTC	Arabic@PAN'17	0.8378	0.8377	0.8385
English	μTC	English@PAN'17	0.8144	0.8143	0.8167
Spanish	μTC	Spanish@PAN'17	0.7756	0.7755	0.7768
Arabic	μTC	Arabic@PAN'17	**0.8378**	**0.8377**	**0.8385**
English	μTC	Arabic@PAN'17	**0.8267**	**0.8266**	**0.8284**
Spanish	μTC	Arabic@PAN'17	**0.7933**	**0.7933**	**0.7943**
Arabic	μTC	English@PAN'17	0.8156	0.8155	0.8162
English	μTC	English@PAN'17	0.8033	0.8026	0.8070
Spanish	μTC	English@PAN'17	0.7733	0.7730	0.7751
Arabic	μTC	Spanish@PAN'17	0.8022	0.8015	0.8040
English	μTC	Spanish@PAN'17	0.8011	0.8005	0.8047
Spanish	μTC	Spanish@PAN'17	0.7756	0.7752	0.7773

4.2 Results for the Image Classification Subtask

The image-based profiling uses our Bag of Visual Words with 5000 centers and $k = 7$ (nearest centroids), with this configuration, our approach produced an accuracy of 0.5691, 0.5468, 0.5900 for Spanish, English, and Arabic languages, respectively. Figure 1 shows several configurations with different parameters and its related scores in the validation set. In this graphs, the y-axis is the value of accuracy, macro-F1, and macro-recall; the x-axis is the number of k nearest centroids used to encode the image into text. These parameters were the result of optimizing for the Spanish language and applying the same for the rest, see Figure 1. While this setup is competitive for the Spanish dataset, it has lower performance in English, and a quite bad in the Arabic dataset, as the array of figures illustrate. However, this multi-lingual analysis was performed after the deadline date, so they were not tested in the TIRA virtual machine. Based on this evidence, it is essential to optimize the parameters to each dataset.

On the other hand, the Rocchio classifier was feed with vectors produced with the text tokenized with character q-grams of length $1, 3, 5$, and 7, the tokens having a frequency lower than 3 in the entire collection were removed, and finally, the TFIDF weighting scheme to encode the bag of tokens into vectors. Please note that this configuration was also selected for the Spanish language and applied for others; it is quite possible that the optimal configuration for each dataset vary; however, this analysis is beyond the scope of this document.

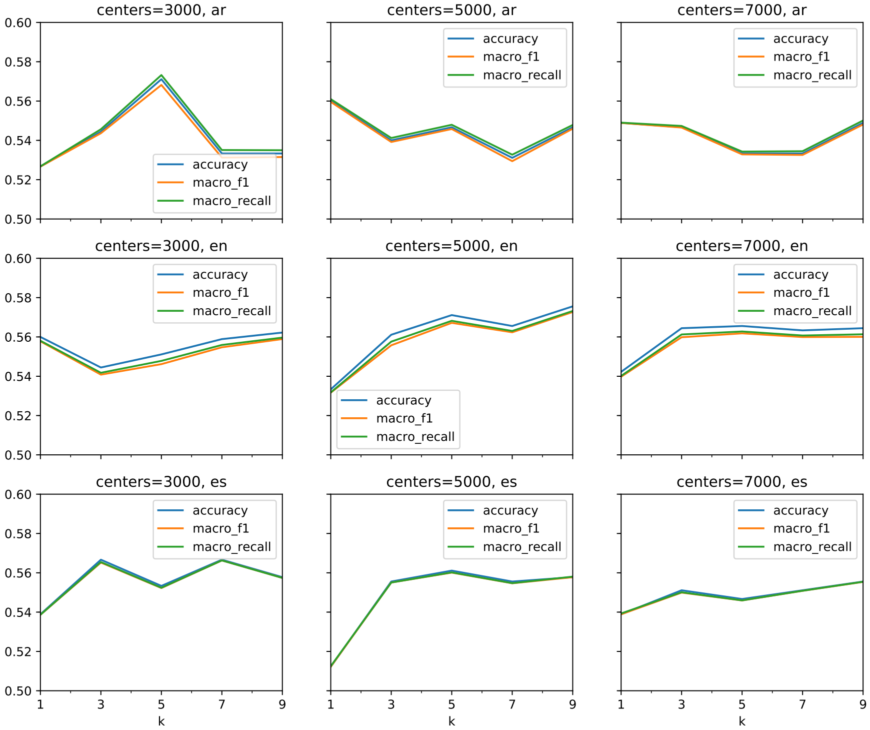

Figure 1. Performance scores of the author profiling task in the image-based subtask.

4.3 Results of the Text and Image Combination

The multi-modal approach uses the analysis of both text and images to predict the gender of the user. We tackled this task using a simple convex combination from the text and images author-profiling models, independently, normalize both predictions and combine them using a convex linear combination using the formulation of §3.2. Table 2 shows the performance of our approach in the evaluation set; also, the table shows the optimized α that maximizes the accuracy performance in the evaluation set.

Table 2. Performance scores of the combination of both text and image predictors.

dataset	α	accuracy	macro-F1	macro-Recall
Arabic	0.99	0.8400	0.8399	0.8408
English	0.95	0.8278	0.8278	0.8293
Spanish	0.925	0.8033	0.8033	0.8042

5 Conclusions

In this notebook, we describe the INGEOTEC's system used to solve the Author Profiling task in PAN'18. We used our MicroTC (μTC) framework to tackled the text classification problem, and for the images, we use a variant of Bag of Visual Words that transform images to text, and not only to histograms like typical BoVW approaches.

We observe that text-based features are dependent of the collection, and its analysis is beyond the scope of this On the other hand, we observed that women tend to share selfies and images with text-content. In case of men, they shared cartoons and humorous images, and landscape photos as well. In our approach, faces and text images are important in this specific problem. In our opinion working on this particular image classification, author profiling based on shared images, task was very difficult, mainly because of the limited amount of images per user. We will be working hard to improve our methods to improve our multi-modal approach to author profiling. Finally, as part of our future work, we will be working in better combination schemes, ad-hoc to the problem particularities.

References

1. Antipov, G., Berrani, S.A., Dugelay, J.L.: Minimalistic cnn-based ensemble model for gender prediction from face images. Pattern Recognition Letters 70, 59 – 65 (2016), http://www.sciencedirect.com/science/article/pii/S0167865515003979
2. Battiti, R., Brunato, M., Mascia, F.: Reactive search and intelligent optimization, vol. 45. Springer Science & Business Media (2008)

3. Bekios-Calfa, J., Buenaposada, J.M., Baumela, L.: Robust gender recognition by exploiting facial attributes dependencies. Pattern Recognition Letters 36, 228 – 234 (2014), http://www.sciencedirect.com/science/article/pii/S0167865513001864
4. Bergstra, J., Bengio, Y.: Random search for hyper-parameter optimization. Journal of Machine Learning Research 13(Feb), 281–305 (2012)
5. Burke, E.K., Kendall, G., et al.: Search methodologies. Springer (2005)
6. Fan, R.E., Chang, K.W., Hsieh, C.J., Wang, X.R., Lin, C.J.: Liblinear: A library for large linear classification. Journal of machine learning research 9(Aug), 1871–1874 (2008)
7. Fatima, M., Hasan, K., Anwar, S., Nawab, R.M.A.: Multilingual author profiling on facebook. Information Processing & Management 53(4), 886 – 904 (2017), http://www.sciencedirect.com/science/article/pii/S0306457316302424
8. Lu, X., Chen, H., Jain, A.K.: Multimodal facial gender and ethnicity identification. In: Zhang, D., Jain, A.K. (eds.) Advances in Biometrics. pp. 554–561. Springer Berlin Heidelberg, Berlin, Heidelberg (2005)
9. Merler, M., Cao, L., Smith, J.R.: You are what you tweet: gender prediction based on semantic analysis of social media images. In: 2015 IEEE International Conference on Multimedia and Expo (ICME). pp. 1–6 (June 2015)
10. Ortega-Mendoza, R.M., L/'opez-Monroy, A.P., Franco-Arcega, A., y G/'omez, M.M.: Emphasizing personal information for author profiling: New approaches for term selection and weighting. Knowledge-Based Systems 145, 169 – 181 (2018), http://www.sciencedirect.com/science/article/pii/S0950705118300224
11. Potthast, M., Rangel, F., Tschuggnall, M., Stamatatos, E., Rosso, P., Stein, B.: Overview of PAN'17: Author Identification, Author Profiling, and Author Obfuscation. In: Jones, G., Lawless, S., Gonzalo, J., Kelly, L., Goeuriot, L., Mandl, T., Cappellato, L., Ferro, N. (eds.) Experimental IR Meets Multilinguality, Multimodality, and Interaction. 8th International Conference of the CLEF Initiative (CLEF 17). Springer, Berlin Heidelberg New York (Sep 2017)
12. 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)
13. Rangel, F., Rosso, P., Potthast, M., Stein, B.: Overview of the 5th Author Profiling Task at PAN 2017: Gender and Language Variety Identification in Twitter. In: Cappellato, L., Ferro, N., Goeuriot, L., Mandl, T. (eds.) Working Notes Papers of the CLEF 2017 Evaluation Labs. CEUR Workshop Proceedings, CLEF and CEUR-WS.org (Sep 2017)
14. Rangel, F., Rosso, P., Potthast, M., Stein, B., Daelemans, W.: Overview of the 3rd author profiling task at pan 2015. In: CLEF (2015)
15. 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)
16. Rocchio, J.J.: Relevance feedback in information retrieval (1971)
17. 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 International Conference of the CLEF Initiative (CLEF 18). Springer, Berlin Heidelberg New York (Sep 2018)

18. Tellez, E.S., Moctezuma, D., Miranda-Jiménez, S., Graff, M.: An automated text categorization framework based on hyperparameter optimization. Knowledge-Based Systems 149, 110 – 123 (2018), https://www.sciencedirect.com/science/article/pii/S0950705118301217
19. Tellez, E.S., Miranda-Jiménez, S., Graff, M., Moctezuma, D.: Gender and language-variety identification with microtc. In: Working Notes of CLEF 2017 - Conference and Labs of the Evaluation Forum, Dublin, Ireland, September 11-14, 2017. (2017), http://ceur-ws.org/Vol-1866/paper_104.pdf
20. Tola, E., Lepetit, V., Fua, P.: Daisy: An efficient dense descriptor applied to wide-baseline stereo. IEEE Transactions on Pattern Analysis and Machine Intelligence 32(5), 815–830 (May 2010)
21. van der Walt, S., Schönberger, J.L., Nunez-Iglesias, J., Boulogne, F., Warner, J.D., Yager, N., Gouillart, E., Yu, T., the scikit-image contributors: scikit-image: image processing in Python. PeerJ 2, e453 (6 2014), http://dx.doi.org/10.7717/peerj.453
22. You, Q., Bhatia, S., Sun, T., Luo, J.: The eyes of the beholder: Gender prediction using images posted in online social networks. In: 2014 IEEE International Conference on Data Mining Workshop. pp. 1026–1030 (Dec 2014)