Multi-Language Neural Network Model with Advance Preprocessor for Gender Classification over Social Media

Notebook for PAN at CLEF 2018

Kashyap Raiyani, Teresa Gonçalves, Paulo Quaresma and Vitor Beires Nogueira

Computer Science Department, University of Évora, Portugal
{kshyp,tcg,pq,vbn}@uevora.pt

Abstract This paper describes approaches for the Author Profiling Shared Task at PAN 2018. The goal was to classify the gender of a Twitter user solely by their tweets. Paper explores a simple and efficient Multi-Language model for gender classification. The approach consists of tweet preprocessing, text representation and classification model construction. The model achieved the best results on the English language with an accuracy of 72.79%; for the Spanish and Arabic languages the accuracy was 72.20% and 64.36%, respectively.

Keywords: Author Profiling, Gender Classification, Twitter, PAN

1 Introduction

The Author Profiling is the identification of demographic features of an author by examining his written text. Recently, it has attracted the attention of research community due to its potential applications in forensic, security, marketing, fake profiles identification on online social networking sites, capturing sender of harassing messages etc. Knowing the profile of an author could be of key importance; for instance, from a forensic linguistics perspective, being able to know what is the linguistic profile of a suspected text message (language used by a certain type of people) and identify characteristics (language as evidence) just by analyzing the text would certainly help considering suspects.

Following this ideology, PAN [26] is a series of scientific events and shared tasks on digital text forensics and stylometry. In PAN 2018 the focus was on Author Identification, Author Profiling, and Author Obfuscation. The Author Identification [11] task was to identify who wrote the document given a document. The subtasks focus on crossdomain authorship attribution applied in fanfiction and on style change detection. Where in Author Profiling [21], the task was to find author's gender whereas text and image might be used as information sources of tweets in English, Spanish and Arabic. Lastly, Author Obfuscation [17], given the document, the tasks were author masking and obfuscation evaluation. Basically, this task works against identification and profiling by automatically paraphrasing a text to obfuscate its author's style.

This paper emphasizes the author profiling shared-task and delivers brief about the recent related work towards gender identification in social media and the technology

and methods used in building systems. Later on, it also shows the shared-task submitted system modeling and its results. It is organized in the following manner: section 2 introduces past research over gender classification and presents a brief introduction to gender classification and its components; section 3 outlines the stylometric data and preprocessing; section 4 describes different data representations; section 5 describes methods and system modeling; section 6 discusses the experimental results obtained. Conclusion and future work are highlighted in Section 7.

2 Related Work

In recent years, efforts have been made by the research community to develop benchmark corpora for author profiling task. The most prominent effort in this regard is the series of PAN competitions on author profiling ([18–23]). The focus of author profiling has settled much on the social media, including languages other than English. This section will cover the previous findings on the author profiling.

The work from Basile *et. al.* [2] reported having the best performance at the PAN 2017 [22]. Their system used a linear support vector machine (SVM) with word unigrams and character 3- to 5-grams as features. Apart from that, they concluded that additional features, including part of speech tags, additional datasets, geographic entities and Twitter handles hurt rather than improve performance.

The approach in Matej *et. al.* [14] has mainly consisted of tweet preprocessing, feature construction, feature weighting and classification model construction with linear SVM, logistic regression, logistic regression bagging, random forest, and XGBoost. The best results were obtained with Logistic Regression and the Bagging & Voting did not improve the results. Here, the main features used were different types of character and word n-grams with supporting features like POS tag sequences, emoji counts, character flood counts, language variety specific word lists and document sentiments. They concluded with that the most difficult part of the task was finding the right features and properly weighing their combination. This approach was able to achieve 2^{nd} highest results among author profiling task at PAN 2017 [22].

Eric *et. al.* [27] used the MicroTC (μTC) framework tool for classification. It follows a simple approach to text classification and converts the problem of text classification to a model selection problem using several simple text transformations, a combination of tokenizers, a term-weighting scheme, and finally, it classifies using a Support Vector Machine. In particular, their system works best when Hill Climbing [7] procedure is applied over the best configuration found by a Random Search [8]. Here, the main idea behind Hill Climbing is to explore the configuration's neighborhood and greedily move to the best neighbor.

The work from Vollenbroek *et. al.* [4] reported having the best performance at the PAN 2016 [23]. They trained SVM linear model using the feature traits like N-grams, starts a capital sentence, capitalized tokens, capital letters, endings with punctuation, punctuation by sentence, average word length and average sentence length, out of dictionary words, vocabulary richness, function words, parts of speech, and emotions.

Pashutan *et. al.* [16] states the extraction of stylistic and lexical features for training a logistic regression model. Here, lexical features such as unigrams and bigrams were

used. Their approach stood first place for gender detection in English and tied for second place in terms of joint accuracy in author profiling PAN 2016 [23].

There are several interesting works on author profiling from the perspective of the common theoretical framework which involves several disciplines such as psychology, (computational) linguistics or even neurology. On the language perspective, features like word unigrams and bigrams, character tetragrams, average spelling error and part of speech (POS) n-grams are considered. In concerned with machine learning algorithms, logistic regression, support vector machine (SVC), multinomial NB, bernoulli NB, ridge classifier, adaBoost classifier and deep learning systems are used. As described, there are a lot of meta-variables that need to be chosen while designing a classification system: the features to be used as input and the machine learning algorithm and its parameter values. Before proceeding with the methodology, some time was taken to understand the data and the requirement of preprocessing.

3 Data Analysis & Preprocessing

The focus of this year's author profile task [21] was on gender identification on Twitter, where text and images were provided as information sources. The languages addressed were English, Spanish, and Arabic. The provided training data set had 3000 Twitter users labeled with gender and, for each user, a total of 100 tweets and 10 images were provided. Authors were grouped by the language of their tweets. The provided data had equal class distribution, meaning that 50% of tweets were from the male and rest 50% were from the female genders. The data had one XML file with 100 tweets for each user. To retrieve the tweets from the XML files, python's ElementTree XML API[1] was used. This API has a parse feature which helps to translate XML tags to a tree and different tags are considered as nodes. By traveling them through root node, users' information was retrieved and stored in one CSV file; the corresponding gender was retrieved from a separate text file. Initially, there were total 3000 XML files with 100 tweets each and after the above-mentioned process, a single CSV file with 300000 row of tweets mapped with the corresponding gender was obtained.

The next subsections describe the text preprocessing steps and preprocessing done for each dataset.

3.1 Preprocessing Steps

This subsections discuss the different preprocessing steps attained to build the input features for the machine learning algorithm.

Merging of Stem Words. Words ending with *ing*, *ious* and *ment* were merged into the corresponding root word. A regular expression was created and used to make this transformation.

[1] `https://docs.python.org/2/library/xml.etree.elementtree.html`

Character Repetition and Stopwords. The extra repetition of characters in a word (for example "tooooo mucccch loveee isss nottt gooooood") were removed using the regular expression followed by the removal of stopwords.

Users in Tweets. The main characteristic of tweets is to have '@' for mentioning user and '#' for tagging. '@' is always followed by the name of the other user. But when you consider the unique users in the provided dataset where 300000 tweets are present, the question comes to the mind is this really necessary to include all the `@usernames`? or should they be replaced by `@user`? with this in mind, @usernames were replaced by word 'user'.

Symbolic Emoji Replacement. In day to day social media writing, people often use the symbolic emojis. This step handles such emojies which are not in form of regular emoji. Table 1 shows the replacements done.

Symbol	Replaced with
:) :) :-) (: (: (-: :')	em_smile
:D : D :-D xD x-D XD X-D	em_laugh
<3 :*	em_love
;-) ;) ;-D ;D (; (-;	em_wink
:-(: (:( ):)-:	em_sad
:,(:'(:"(	em_cry
X(>:-(>:(X-(	em_angry

Table 1. Symbolic Emoji Replacement

Segmentation. One challenge was to identify the meaning of the # or hashtag. There are couple of tools available like nltk Segmentation [13] and Ekphrasis Segmentation [3] that can be used to segment and split hashtags.Here, Ekphrasis was used for segmentation. Apart from that, another tool present in the Ekphrasis library was used to replace a particular segment of the tweets with its corresponding type. The types considered were 'url', 'email', 'percent', 'money', 'phone', 'time', 'date' and 'number'. This library uses the regular expressions to normalize text segments.

Communication Abbreviations. Another preprocessing step was to remove day to day communication abbreviations. Like consider two sentences "I am in love with you" and "im in luv wid u". Both the sentences have the same meaning and might be written by the same author but the machine will see them as two different representations. The motivation is to reduce the confusion/possibility for the machine learning model. Table 2 presents the preprocessing done over a day to day communication abbreviations.

3.2 Data Preprocessing

For the Arabic dataset no preprocessing was done but for the English dataset, all the above mentioned preprocessing approaches were performed before the text representation to be presented to the machine learning algorithm.

Word	Replaced with
app, wil, im, al, sx, u	application, will, i am, all, sex, you
r, y, hv, c, bcz or coz	are, why, have, see, because
ppl or pepl, nd, hw	people, and, how

Table 2. Abbreviation Preprocessor

One of the most common things that Spanish native speakers do when they text is omitting some of the letters in words and using just the endings. For example, the "es" sound at the beginning of words like "estoy", "estás" or "estamos" will be dropped, leaving with "toy", "tás" and "tamos" instead. Dropping the "d" is widespread orally, and it is also used when texting. For example, "cansado" is "cansao" and "todo" is "to" or "too." If a friend texts "toy cansao" it means "estoy cansado" or "too tá listo" it means "todo está listo." With this much versatility in the language, it was hard to design the preprocessor without having Spanish language background. For that matter, only symbolic emojis were replaced (see table 1) followed by the removal of the Spanish stopwords. Here, merging of stem words was not done.

4 Text Representation

This section describes the text representation before passed to the machine learning algorithm.

A sequential model can be designed for text classification where the text sequence is fed to the machine learning algorithm[2]. Nonetheless, these models only use numerical data and therefore a conversion was needed which involves two subprocesses: Integer Encoding and One-Hot Encoding [12].

Integer Encoding. Here, each unique word/text value is assigned an integer value for example, "red" is 1, "green" is 2, and "blue" is 3. This process is known as label encoding or an integer encoding and is easily reversible. The integer values have a natural ordered relationship between each other and machine learning algorithms may be able to understand and harness this relationship. For example, ordinal variables like the "place" example above would be a good example where a label encoding would be sufficient. But for sentences where no such ordinal relationship exists, the integer encoding is not enough which leads to the next encoding One-Hot encoding.

$$\begin{bmatrix} red & green & blue \\ 1 & 2 & 3 \end{bmatrix}$$

One-Hot Encoding. Here, the integer encoded variable is removed and a new binary variable is added for each unique integer value. In the "color" variable example, there

[2] `https://keras.io/getting-started/sequential-model-guide/`

are 3 words and therefore 3 binary variables are needed. For color, "1" value is placed in the binary variable and "0" values for rest of the colors. For example:

$$\begin{bmatrix} red & green & blue \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Representation. Both for English and Spanish, a word dictionary was created in which, all the unique words were indexed; the index was then used as the word id. Using Integer Encoding and One-Hot Encoding concepts, these ids were portrayed as integers first and then as a binary matrix preserving the word order of the sentences.

For better understanding, assume the English dictionary {"country": 1, "very": 2, "I": 3, "love": 4, "my": 5}. Considering the Figure 1, both sentences use the same word set but their sentence ordering is preserved using the dictionary index and one-hot approach.

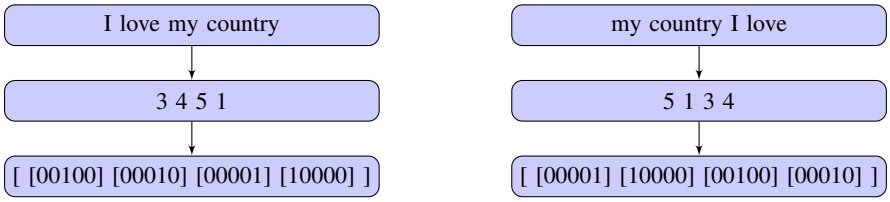

Figure 1. English Text Converstion

The given Arabic dataset was smaller in size compared to the English and Spanish ones; only 1500 XML files were provided. Here, XML files were not trasnformed to a CSV file; rather a single text file was created where each line contained a single tweet from the user. These lines were prefixed with author's gender. As a text representation, a character unigram representation was used.

5 System Modeling

For the English and Spanish languages, a dense Neural Network was considered; for Arabic, FastText [9] library was used. FastText is an open-source library developed by Facebook Open Source for researchers to learn and test text classification problems. It provides word vectors for 157 languages and supervised models for 8 datasets using a character level n-gram approach. For research and development porpose, Keras [5] was used as front-end and Tensorflow [1] as back-end.

The next sub-sections present the system architecture design.

5.1 Dense Neural Network

After experimenting with different methods and machine learning algorithms, a simple Dense architecture was used to create the final model for submission. The figure 2 represents the architecture layers of the Neural Network.

After several iterations, the number of layers for the architecture was set to three. The 1st hidden layer is having 1024 nodes which take the input from the input layer and the 2nd layer had 512 nodes and the last layer had 256 nodes.

Inbetween each layers, mathamatical activation functions like Relu, Sigmoid and Softmax were used. The ordering of this activation function had very high impact on the results.

The table 3 presents the experiment parameters.

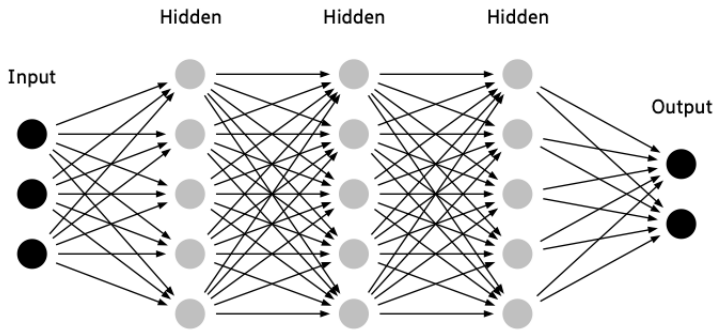

Figure 2. System Architecture

Parameter	Value
Maximum Number of Words	2000
Validation Split Ratio	0.2
Epochs	2
Batch Size	32
Optimizer	Adam

Table 3. Experiment Parameters

5.2 FastText

According to Joulin *et. al.* [10], a simple and efficient baseline for sentence classification is to represent sentences as a bag of words (BoW) and train a linear classifier, e.g., a logistic regression or an SVM [25, 28]. However, linear classifiers do not share parameters among features and classes. For each example, y is a scalar that takes binary values ($\hat{y} \in [0, 1]$), while x is a vector of length N (where N is the number of features). Here, the the output is a linear combination of the input features (i.e. a weighted sum of these features plus the biases).

$$\hat{y} = \left(\sum_i^N (w_i \cdot x_i) + b\right) \quad or \quad (\mathbf{w} \cdot \mathbf{x} + b)$$

where x and w are vectors of length N and the product $x \cdot w$ produces a scalar. As can be seen, a separate weight w_i for each input feature x_i is considered and these weights are independent by all means. This shows that there is no parameter sharing among features for linear classifiers. This possibly limits their generalization in the context of large output space where some classes have very few examples. Common solutions to this problem are to factorize the linear classifier into low-rank matrices [6, 15] or to use multilayer neural networks [24, 29]

Figure 3 shows a simple linear model with rank constraint architecture which was used for Arabic text classification (with the help of FastText).

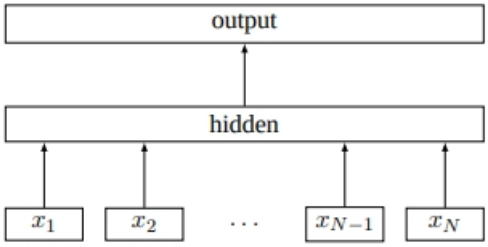

Figure 3. Model architecture of FastText for a sentence with N ngram features $x_1, \ldots, x_N$. The features are embedded and averaged to form the hidden variable

An Architecture similar to the cbow model of Mikolov *et. al.* [15], where the middle word is replaced by a label, here, the first weight matrix is taken as a look-up table over the words and then the word representations are averaged into a text representation, which is in turn fed to a linear classifier. The text representation is a hidden variable which can be potentially be reused. The softmax function was used to compute the probability distribution over the predefined classes. For a set of N documents, this leads to minimizing the negative loglikelihood over the classes

$$-\frac{1}{N}\sum_{n=1}^{N} y_n log(f(BAx_n))$$

where x_n is the normalized bag of features of the nth document, y_n the label, A and B the weight matrices. This model has trained asynchronously on multiple CPUs using stochastic gradient descent and a linearly decaying learning rate.

The parameters of training were 150000 sentences as an input(or N), learning rate of 0.25, hidden value of 100(or h), 40 epochs and softmax loss function. With this, the system was able to learn 99.98% on a train set.

6 Results and Discussion

Table 4 presents the results provided by the PAN 2018 organizing committee for the systems described in the previous section.

Language	Accuracy(%)
English	72.79
Spanish	64.36
Arabic	72.20

Table 4. Results for PAN 2018 Author Profiling Task

From the Table 4, one can say that the English and Arabic systems performed better than the Spanish one. This difference can probably be explained by the fact that not all described preprocessing filters steps were applied to the Spanish system: only the simple regular expression transformation, stopwords removal and emoji normalization were done. Apart from this, given the amount of dataset, with provided hyper parameters, English & Spanish model reached to the saturation point of the designed model.

7 Conclusion and Future Work

This paper presents a system for multi-language author profiling task for gender classification and presents the findings from the related work. Further, it concludes that, for the current author profiling task, a seemingly simple system using fully connected architecture proves beyond average performance. Paper also described the preprocessing techniques used, the methodology of approach and the conducted experiments.

As mentioned in the previous section 6, models reached its saturation point for the design. This leads to the one limitation of this approach. Proposed systems are only suitable when having a small dataset; with the large dataset, the dictionary index will be huge and the system won't be able to convert all the words to a binary matrix.

In this model words that are not found in the dictionary are omitted. As future work, and to address this issue, a neighboring or similar word could be used; here each word would be added to a list from which neighboring/similar word would be founded. This will help in categorizing unseen words to the model. Thus, all the words are included and there is no omission of words. Another line of future work can be to include semantic meaning from the sentences, extracting ontological features either by Part of Speech (POS) tagging or Entity Extraction (EE).

Acknowledgement

This research was supported by AGATHA (Intelligent analysis system of open information sources for surveillance/crime control) project and by the University of Évora.

References

1. Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., Kudlur, M., Levenberg, J., Monga, R., Moore, S., Murray, D.G., Steiner, B., Tucker, P., Vasudevan, V., Warden, P., Wicke, M., Yu, Y., Zheng, X.:

Tensorflow: A system for large-scale machine learning. In: Proceedings of the 12th USENIX Conference on Operating Systems Design and Implementation. pp. 265–283. OSDI'16, USENIX Association, Berkeley, CA, USA (2016), http://dl.acm.org/citation.cfm?id=3026877.3026899
2. 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: Cappellato et al. [22], http://ceur-ws.org/Vol-1866/
3. Baziotis, C., Pelekis, N., Doulkeridis, C.: Datastories at semeval-2017 task 4: Deep lstm with attention for message-level and topic-based sentiment analysis. In: Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017). pp. 747–754. Association for Computational Linguistics, Vancouver, Canada (August 2017)
4. Busger op Vollenbroek, M., Carlotto, T., Kreutz, T., Medvedeva, M., Pool, C., Bjerva, J., Haagsma, H., Nissim, M.: GronUP: Groningen User Profiling—Notebook for PAN at CLEF 2016. In: Balog et al. [23], http://ceur-ws.org/Vol-1609/
5. Chollet, F., et al.: Keras. https://keras.io (2015)
6. H., S.: Dimensions of meaning. In: Proceedings of the 1992 ACM/IEEE Conference on Supercomputing. pp. 787–796. Supercomputing '92, IEEE Computer Society Press, Los Alamitos, CA, USA (1992), http://dl.acm.org/citation.cfm?id=147877.148132
7. Jacobson, S.H., Yücesan, E.: Analyzing the performance of generalized hill climbing algorithms. Journal of Heuristics 10(4), 387–405 (Jul 2004), https://doi.org/10.1023/B:HEUR.0000034712.48917.a9
8. James, B., Yoshua, B.: Random search for hyper-parameter optimization. J. Mach. Learn. Res. 13, 281–305 (Feb 2012), http://dl.acm.org/citation.cfm?id=2188385.2188395
9. Joulin, A., Grave, E., Bojanowski, P., Mikolov, T.: Bag of tricks for efficient text classification. arXiv preprint arXiv:1607.01759 (2016)
10. Joulin, A., Grave, E., Bojanowski, P., Mikolov, T.: Bag of tricks for efficient text classification. CoRR abs/1607.01759 (2016), http://arxiv.org/abs/1607.01759
11. Kestemont, M., Tschugnall, M., Stamatatos, E., Daelemans, W., Specht, G., Stein, B., Potthast, M.: Overview of the Author Identification Task at PAN-2018: Cross-domain Authorship Attribution and Style Change Detection. 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)
12. Lantz, B.: Machine Learning with R. Packt Publishing (2013)
13. Loper, E., Bird, S.: Nltk: The natural language toolkit. In: Proceedings of the ACL-02 Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics - Volume 1. pp. 63–70. ETMTNLP '02, Association for Computational Linguistics, Stroudsburg, PA, USA (2002), https://doi.org/10.3115/1118108.1118117
14. Martinc, M., Škrjanec, I., Zupan, K., Pollak, S.: PAN 2017: Author Profiling - Gender and Language Variety Prediction—Notebook for PAN at CLEF 2017. In: Cappellato et al. [22], http://ceur-ws.org/Vol-1866/
15. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. CoRR abs/1301.3781 (2013), http://arxiv.org/abs/1301.3781
16. Modaresi, P., Liebeck, M., Conrad, S.: Exploring the Effects of Cross-Genre Machine Learning for Author Profiling in PAN 2016—Notebook for PAN at CLEF 2016. In: Balog et al. [23], http://ceur-ws.org/Vol-1609/

17. Potthast, M., Hagen, M., Schremmer, F., Stein, B.: Overview of the Author Obfuscation Task at PAN 2018: A New Approach to Measuring Safety. 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)
18. Rangel, F., Celli, F., Rosso, P., Potthast, M., Stein, B., Daelemans, W.: Overview of the 3rd Author Profiling Task at PAN 2015. In: Cappellato et al. [18]
19. Rangel, F., Rosso, P., Chugur, I., Potthast, M., Trenkmann, M., Stein, B., Verhoeven, B., Daelemans, W.: Overview of the 2nd Author Profiling Task at PAN 2014. In: Cappellato et al. [19]
20. Rangel, F., Rosso, P., Koppel, M., Stamatatos, E., Inches, G.: Overview of the Author Profiling Task at PAN 2013. In: Forner et al. [20]
21. 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)
22. Rangel Pardo, 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, vol. 1866. CLEF and CEUR-WS.org (Sep 2017), http://ceur-ws.org/Vol-1866/
23. Rangel Pardo, F., Rosso, P., Verhoeven, B., Daelemans, W., Potthast, M., Stein, B.: Overview of the 4th Author Profiling Task at PAN 2016: Cross-Genre Evaluations. In: Working Notes Papers of the CLEF 2016 Evaluation Labs. CEUR Workshop Proceedings, vol. 1609. CLEF and CEUR-WS.org (Sep 2016), http://ceur-ws.org/Vol-1609/
24. Ronan, C., Jason, W.: A unified architecture for natural language processing: Deep neural networks with multitask learning. In: Proceedings of the 25th International Conference on Machine Learning. pp. 160–167. ICML '08, ACM, New York, NY, USA (2008), http://doi.acm.org/10.1145/1390156.1390177
25. Rong-En, F., Kai-Wei, C., Cho-Jui, H., Xiang-Rui, W., Chih-Jen, L.: Liblinear: A library for large linear classification. J. Mach. Learn. Res. 9, 1871–1874 (Jun 2008), http://dl.acm.org/citation.cfm?id=1390681.1442794
26. 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)
27. Tellez, E., Miranda-Jiménez, S., Graff, M., Moctezuma, D.: Gender and language-variety Identification with MicroTC—Notebook for PAN at CLEF 2017. In: Cappellato et al. [22], http://ceur-ws.org/Vol-1866/
28. Thorsten, J.: Text categorization with support vector machines: Learning with many relevant features. In: Claire, N., Céline, R. (eds.) Machine Learning: ECML-98. pp. 137–142. Springer Berlin Heidelberg, Berlin, Heidelberg (1998)
29. Zhang, X., Zhao, J.J., LeCun, Y.: Character-level convolutional networks for text classification. CoRR abs/1509.01626 (2015), http://arxiv.org/abs/1509.01626