Gender and language-variety identification with MicroTC

Notebook for PAN at CLEF 2017

Eric S. Tellez[1], Sabino Miranda-Jiménez[1], Mario Graff[1], and Daniela Moctezuma[2]

[1] CONACyT-INFOTEC Centro de Investigación e Innovación en Tecnologías de la Información y Comunicación, México
{eric.tellez, sabino.miranda, mario.graff}@infotec.mx
[2] CONACyT-CentroGEO Centro de Investigación en Geografía y Geomática "Ing. Jorge L. Tamayo" A.C., México
dmoctezuma@centrogeo.edu.mx

Abstract In this notebook, we describe our approach to cope with the Author Profiling task on PAN17 which consists of both gender and language identification for Twitter's users. We used our MicroTC (μTC) framework as the primary tool to create our classifiers. μTC follows a simple approach to text classification; it 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. Our approach reaches accuracies of 0.7838, 0.8054, 0.7957, and 0.8538, for gender identification; and for language variety, it achieves 0.8275, 0.9004, 0.9554, and 0.9850. All these, for Arabic, English, Spanish, and Portuguese languages, respectively.

1 Introduction

Recently, forensic text analysis about originality, authorship, and reliability has attracted a lot of attention by researchers and practitioners because of practical applications in security and marketing [19]. In this context, author profiling is an important task of PAN@CLEF forum that focuses on analyzing some characteristics of the author (profiling aspects) based on the written author's text, such as gender, age, political preferences, personality, language variety, among others [14].

Generally speaking, author profiling task is tackled using, mainly, machine-learning approaches, i.e., models, for predicting profiling aspects, are built considering a set of general features that represent different categories of authors, e.g., gender, range age, and language variety, among others [16].

PAN forum 2017[3] provides a dataset of tweets for training and test the performance of each participating system. In this edition, the profiling aspects to

[3] http://pan.webis.de/clef17/pan17-web/author-profiling.html

be analyzed are gender and language of Twitter's users. The corpus is annotated with authors' gender and their particular variation of their mother tongue that includes Arabic, English, Spanish, and Portuguese.

Our approach is language independent, that is, we deliberately avoid the use of linguistic procedures such as part-of-speech tagging, lemmatization, or stemming. In the same way, linguistic resources, like lexicons and WordNet-based, are disallowed. In contrast, we take advantage of multiple tokenizers, an entropy-based term-weighting scheme, and an SVM classifier, see Section 3 for details.

The rest of the paper is organized as follows. Section 2 presents few of the gender, age, language, and region identification related works, and Section 3 describes our system and the general approach to model the problem. Section 4 detail the experimental methodology and the achieved results. Finally, conclusions and future work are given in Section 5.

2 Related work

Author profiling is a repetitive and important task in PAN contest since 2013 [16]. Before 2017 edition, only age and gender classification tasks were considered [17,18]. This year, the PAN considers the region aspect while removes the age identification subtask from the competition [14].

Several works have been proposed to solve age and gender identification subtasks. Agrawal & Gonçalves [1] use a combination of classifiers along with a model based on user's activities to predict the profile of the unknown users. The TFIDF representation was employed, and a dimension reduction was performed in this matrix. The authors use Naive Bayes and Linear SVM as classifiers.

With the purpose to find the differences between writing styles of males and females in different age groups, the usage of several stylometric features is considered in [4]. Another stylometric approach was presented in [5] where two groups of features were considered, trigrams and complementary-weighted *Second Order Attributes*. An SVM classifier is used in the classification step. A combination of features based on word n-grams, sentences starting with capital letters, finish the sentences with a dot, emoticons, word's length and sentence's length is also used along with grammatical aspects are explored in [23].

Lopez-Monroy et al. [12] propose a representation for documents that capture discriminative and subprofile-specific information of terms. Under the proposed representation, terms are represented in a vector space that captures discriminative information. On the other hand, more traditional representations, like TFIDF, are broadly employed in the author's profiling literature, that is the case of [8], [22], and [13]. Classification ensembles are also frequently used; for instance, [24] generate several classifiers using sets of features such as word n-gram, character n-gram, and part-of-speech n-gram features.

Language variety identification is a new subtask introduced in PAN17 that consists in determining the specific variation of the native language of authors' text [11,10]. Another approach to region classification is presented in [7] where

twitter geolocation and regional classification was conducted through sparse coding and dictionary learning. Another region prediction approach based on Modified Adsorption, removing "celebrity" nodes and analyzing a graph model propagation is proposed in [15].

3 System description

MicroTC (μTC) is a generic framework for text classification task, i.e., it works regardless of both domain and language particularities. μTC is an extension of our previous work on sentiment analysis, see [20]. A full description of μTC can be found in [21]. The core idea behind μTC is to tackle a text classification task by selecting an appropriate configuration from a set of different text transformations techniques, tokenizers, and several weighting schemes, using as a classifier a Support Vector Machine (SVM) with linear kernel. In some sense, the text classification problem is transformed into *hyper-parameter optimization*, also known as model selection.

3.1 About μTC

Briefly, μTC contains the following parts: i) a list of functions that normalize and transform the input text to the input of tokenizers (preprocessing), ii) a set of tokenizer functions that transform the filtered text into a multiset of tokens, iii) a function that generates weighted vectors from the multiset of tokens; and finally, iv) a classifier that knows how to assign a label to a given vector.

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 scheme to weight the coordinates of the vector. It contains a number of frequency filters that were deactivated for this contribution, see [21] for more details.

iv. **Classifier** We decide to use a singleton set populated with an SVM with a linear kernel. It is well known that SVM performs excellently for very 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. [9].

3.2 Modeling users

We select to model a user using all its tweets, that is, an user u is a collection of small texts $u = \{t_1, \ldots, t_n\}$. For each text, we apply the preprocessing step and tokenizers, then we create a multiset from the union of all multisets in u. After this, a vector $\boldsymbol{u}$ is created using a term weighting scheme. Thus, we modeled each user as a high dimensional sparse vector. For instance, since we do not remove any kind of terms, and in fact we promote the usage of combinations of tokenizers, the user's vectors can contain millions of coordinates, and thousand non-zero entries.

The weighting schemes for this modeling are described in the following paragraphs. We also introduce *entropy+b*, a new weighting scheme introduced in this notebook designed for classification tasks. In the following paragraphs we describe in detail the weighting schemes used in the experimental section.

The simpler scheme corresponds to freq, and it is defined as the term frequency of each term per user; we name it $\mathsf{freq}_{\mathsf{usr}}$ to avoid confusion with other functions. TFIDF is the product of TF and IDF where TF is the normalized frequency of a user's term, and IDF is the inverse document frequency defined as the logarithm of the inverse of the probability that a term occurs in the whole collection of users, more precisely,

$$\mathsf{TF}(w, \mathsf{usr}) = \frac{\mathsf{freq}_{\mathsf{usr}}(w)}{\max_{w \in \mathsf{usr}} \{\mathsf{freq}_{\mathsf{usr}}(w)\}},$$

and

$$\mathsf{IDF}(w) = \log \frac{N}{|\{\mathsf{usr} \mid \mathsf{freq}_{\mathsf{usr}}(w) > 0\}|},$$

where N is the size of the training collection, i.e., the number of users. It is common to add 1 to the denominator expression to avoid numerical problems.

In this notebook, we introduce the *entropy+b* term-weighting that considers that each term is represented by a distribution over the available classes. Instead of using the raw probabilities per class, we weight each term with the Entropy+b function, 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,

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

Here, freq_c denotes the frequency of the given term in the class c. The idea behind $\mathsf{entropy}_b(w)$ is to weight each term using the entropy of the underlying distribution in a way that large entropy values (terms uniformly distributed along all classes) have a low weight while terms being skewed to some class are close to $\log |C|$. The parameter b is introduced to *absorb* the possible *noise* that occurs in low populated terms.

3.3 About the model selection

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, The procedure is repeated until some condition is achieved, like the impossibility of improve the score function, or when a maximum number of iterations is reached. There are several meta-heuristics to solve combinatorial optimization problems, the proper survey of the area is beyond the scope of this notebook; however, the interested reader is referred to [20,21,6,2].

In particular, μTC uses two types of meta-heuristics, *Random Search* [3] and *Hill Climbing* [6,2] 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. We also add memory to avoid a configuration to be evaluated twice[4].

4 Experiments and results

The experiments with the training set were run in an Intel(R) Xeon(R) CPU E5-2640 v3 @ 2.60GHz with 32 threads and 192 GiB of RAM running CentOS 7.1 Linux. The gold-standard were evaluated in the TIRA platform using a virtual machine with 4GiB of RAM and one core. We implemented μTC[5] on Python.

We partition the full training dataset into two smaller sets, a new training set containing 30% of the users, and a validation set with the resting 70%. The

[4] In principle, this is similar to Tabu search; however, our implementation is simpler than a typical implementation of Tabu search.

[5] Available under Apache 2 license at https://github.com/INGEOTEC/microTC

partition where selected to ensure the generalization of our scheme. On the new training set, from now on just *training* set, we run μTC using random search and hill climbing to perform the hyper-parameter optimization. Random search was allowed to select 32 random configurations. On the other hand, Hill-climbing starts with the best configuration found by random search; the procedure was left to finish its optimization process. 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 in the validation set.

Table 1. Performance of our approaches for gender using $30-70\%$ partition for training and test datasets.

name	**macro-recall**	**macro-f1**	**accuracy**	**improvement**
Arabic				
μTC-FREQ	0.7365	0.7355	0.7369	-
μTC-TFIDF	0.7190	0.7149	0.7169	$\downarrow$2.71%
μTC-entropy+0	0.7030	0.7009	0.7025	$\downarrow$4.66%
μTC-entropy+3	0.7591	0.7583	0.7588	$\uparrow$2.97%
μTC-entropy+10	0.7577	0.7573	0.7575	$\uparrow$2.80%
μTC-entropy+30	0.7460	0.7450	0.7456	$\uparrow$1.19%
μTC-entropy+100	0.7259	0.7252	0.7256	$\downarrow$1.53%
English				
μTC-FREQ	0.7789	0.7787	0.7788	-
μTC-TFIDF	0.7750	0.7738	0.7740	$\downarrow$0.61%
μTC-entropy+0	0.7626	0.7624	0.7625	$\downarrow$2.09%
μTC-entropy+3	0.7897	0.7895	0.7896	$\uparrow$1.39%
μTC-entropy+10	0.7788	0.7787	0.7788	0.0%
μTC-entropy+30	0.7725	0.7725	0.7725	$\downarrow$0.80%
μTC-entropy+100	0.7722	0.7721	0.7721	$\downarrow$0.86%
Spanish				
μTC-FREQ	0.7364	0.7364	0.7364	-
μTC-TFIDF	0.7304	0.7294	0.7296	$\downarrow$0.92%
μTC-entropy+0	0.7105	0.7092	0.7104	$\downarrow$3.54%
μTC-entropy+3	0.7533	0.7527	0.7532	$\uparrow$2.28%
μTC-entropy+10	0.7433	0.7430	0.7432	$\uparrow$0.92%
μTC-entropy+30	0.7415	0.7411	0.7414	$\uparrow$0.68%
μTC-entropy+100	0.7368	0.7366	0.7368	$\uparrow$0.05%
Portuguese				
μTC-FREQ	0.8043	0.8034	0.8038	-
μTC-TFIDF	0.7786	0.7786	0.7788	$\downarrow$3.11%
μTC-entropy+0	0.8406	0.8395	0.8400	$\uparrow$4.51%
μTC-entropy+3	0.8440	0.8437	0.8438	$\uparrow$4.98%
μTC-entropy+10	0.8464	0.8462	0.8463	$\uparrow$5.29%
μTC-entropy+30	0.8377	0.8375	0.8375	$\uparrow$4.20%
μTC-entropy+100	0.8240	0.8237	0.8238	$\uparrow$2.49%

Table 1 shows the performance of μTC for gender identification. In particular, we show macro-recall, macro-f1, and accuracy scores. We show three different term-weighting schemes, detailed in §3. We select the FREQ scheme to

describe the improvement of each scheme. The FREQ and $TFIDF$ schemes are implemented in μTC; for entropy+b, we show the performance for five different values of b. Table 1 indicates that $TFIDF$ performs poorly as compared with FREQ. Entropy+b illustrates the dependency of b, showing better performances for small b values, except $b = 0$ which has a poor performance for gender identification. The table shows that $b = 3$ and $b = 10$ performs much better than the rest of the classifiers. Between entropy+3 and entropy+10, the first one performs better; however, entropy+3 was evaluated after the deadline of the second run. Therefore, entropy+10 was used to classify the gold standard, see Table 3.

Table 2. Performance of our approaches for language variety using $30 - 70\%$ partition for training and test datasets.

name	**macro-recall**	**macro-f1**	**accuracy**	**improvement**
Arabic				
μTC-FREQ	0.7577	0.7594	0.7581	-
μTC-TFIDF	0.7488	0.7488	0.7488	$\downarrow$1.24%
μTC-entropy+0	0.8088	0.8098	0.8094	$\uparrow$6.76%
μTC-entropy+3	0.8111	0.8118	0.8113	$\uparrow$7.01%
μTC-entropy+10	0.8039	0.8047	0.8044	$\uparrow$6.10%
μTC-entropy+30	0.8164	0.8169	0.8169	$\uparrow$7.75%
μTC-entropy+100	0.8070	0.8073	0.8075	$\uparrow$6.51%
English				
μTC-FREQ	0.7834	0.7839	0.7833	-
μTC-TFIDF	0.7960	0.7957	0.7960	$\uparrow$1.62 %
μTC-entropy+0	0.8901	0.8902	0.8900	$\uparrow$13.62%
μTC-entropy+3	0.8918	0.8921	0.8917	$\uparrow$13.83%
μTC-entropy+10	0.8784	0.8787	0.8783	$\uparrow$12.13%
μTC-entropy+30	0.8683	0.8687	0.8683	$\uparrow$10.85%
μTC-entropy+100	0.8645	0.8649	0.8646	$\uparrow$10.37%
Spanish				
μTC-FREQ	0.9020	0.9022	0.9018	-
μTC-TFIDF	0.8948	0.8947	0.8954	$\downarrow$0.71%
μTC-entropy+0	0.9573	0.9573	0.9571	$\uparrow$6.14%
μTC-entropy+3	0.9537	0.9537	0.9536	$\uparrow$5.74%
μTC-entropy+10	0.9437	0.9437	0.9436	$\uparrow$4.63%
μTC-entropy+30	0.9272	0.9269	0.9268	$\uparrow$2.77%
μTC-entropy+100	0.9109	0.9109	0.9107	$\uparrow$0.99%
Portuguese				
μTC-FREQ	0.9815	0.9812	0.9813	-
μTC-TFIDF	0.9737	0.9737	0.9738	$\downarrow$0.76%
μTC-entropy+0	0.9852	0.9850	0.9850	$\uparrow$0.38%
μTC-entropy+3	0.9901	0.9900	0.9900	$\uparrow$0.89%
μTC-entropy+10	0.9852	0.9850	0.9850	$\uparrow$0.38%
μTC-entropy+30	0.9876	0.9875	0.9875	$\uparrow$0.64%
μTC-entropy+100	0.9852	0.9850	0.9850	$\uparrow$0.38%

Table 2 shows the performance of our systems in the language variety task. As before, we use FREQ as the baseline method. In this task, FREQ also performs

Table 3. Performance of our approaches for language's variety in the official PAN17's gold-standard using μTC with two different term-weighting schemes.

name	language	gender accuracy	variety accuracy	joint accuracy
μTC-FREQ	ar	0.7569	0.7925	0.6125
μTC-entropy+10	ar	0.7838	0.8275	0.6713
μTC-FREQ	en	0.7938	0.8388	0.6704
μTC-entropy+10	en	0.8054	0.9004	0.7267
μTC-FREQ	es	0.7975	0.9364	0.7518
μTC-entropy+10	es	0.7957	0.9554	0.7621
μTC-FREQ	pt	0.8038	0.9750	0.7850
μTC-entropy+10	pt	0.8538	0.9850	0.8425

better than TFIDF, excepting for English; both approaches are part of the μTC tool. The entropy+b scheme is much better for almost any of the presented b's, even for $b = 0$. As in the gender identification task, the smaller values of b perform better than larger values, achieving the best performance when $b = 3$. Nonetheless, we used entropy+10 to classify the gold standard because the deadline hit us.

The official performances on the PAN17 gold standard are shown in Table 3. We send our baseline based on the FREQ weighting scheme and the profiler based on entropy+10. The table indicates the accuracy for gender and variety tasks, as well for the joint accuracy (the same example was correctly predicted in both tasks). As predicted in Tables 1 and 2, entropy+10 has a better performance than FREQ, in some languages by a large margin, e.g., close to five percentual points for Arabic, and six percentual points for Portuguese.

5 Conclusions

In this notebook, we describe the INGEOTEC's system used to solve the Author Profiling task in PAN17. We used our MicroTC (μTC) framework [21] as the primary tool to create our classifiers. μTC follows a simple approach to text classification; it 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 an SVM classifier. It is designed to tackle text-classification problems in an agnostic way, being both domain and language independent.

To effectively tackle the task, we introduce a new term-weighting scheme based on the distributional representation of each term and the entropy over that distribution. We call it *entropy+b*. More work is needed to characterize the new weighting scheme yet it demonstrated to be superior to raw term frequency and TFIDF, at least, for the Author Profiling task and our μTC framework.

Acknowledgements

We like to thank the PAN organizers, in particular to Francisco Rangel and Martin Potthast for their kind and quick response to our questions and requests.

References

1. Agrawal, M., Gonçalves, T.: Age and gender identification using stacking for classification. notebook for pan at clef 2016 (2016)
2. Battiti, R., Brunato, M., Mascia, F.: Reactive search and intelligent optimization, vol. 45. Springer Science & Business Media (2008)
3. Bergstra, J., Bengio, Y.: Random search for hyper-parameter optimization. Journal of Machine Learning Research 13(Feb), 281–305 (2012)
4. Bilan, I., Zhekova, D.: Caps: A cross-genre author profiling system (2016)
5. Bougiatiotis, K., Krithara, A.: Author profiling using complementary second order attributes and stylometric features (2016)
6. Burke, E.K., Kendall, G., et al.: Search methodologies. Springer (2005)
7. Cha, M., Gwon, Y., Kung, H.: Twitter geolocation and regional classification via sparse coding. In: ICWSM. pp. 582–585 (2015)
8. Dichiu, D., Rancea, I.: Using machine learning algorithms for author profiling in social media (2016)
9. 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)
10. Francisco Rangel, Marc Franco-Salvador, P.R.: A low dimensionality representation for language variety identification. In: In Postproc. 17th Int. Conf. on Comput. Linguistics and Intelligent Text Processing, CICLing-2016. Springer-Verlag (2017)
11. Franco-Salvador, M., Rangel, F., Rosso, P., Taulé, M., Martít, M.A.: Language variety identification using distributed representations of words and documents. In: International Conference of the Cross-Language Evaluation Forum for European Languages. pp. 28–40. Springer (2015)
12. López-Monroy, A.P., y Gómez, M.M., Escalante, H.J., Villaseñor-Pineda, L., Stamatatos, E.: Discriminative subprofile-specific representations for author profiling in social media. Knowledge-Based Systems 89, 134 – 147 (2015), http://www.sciencedirect.com/science/article/pii/S0950705115002427
13. Markov, I., Gómez-Adorno, H., Sidorov, G., Gelbukh, A.: Adapting cross-genre author profiling to language and corpus. In: Proceedings of the CLEF. pp. 947–955 (2016)
14. 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)
15. Rahimi, A., Cohn, T., Baldwin, T.: Twitter user geolocation using a unified text and network prediction model. arXiv preprint arXiv:1506.08259 (2015)

16. 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)
17. Rangel, F., Rosso, P., Potthast, M., Stein, B., Daelemans, W.: Overview of the 3rd author profiling task at pan 2015. In: CLEF (2015)
18. 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)
19. Rosso, P., Rangel, F., Potthast, M., Stamatatos, E., Tschuggnall, M., Stein, B.: Overview of PAN'16—New Challenges for Authorship Analysis: Cross-genre Profiling, Clustering, Diarization, and Obfuscation. In: Fuhr, N., Quaresma, P., Larsen, B., Gonçalves, T., Balog, K., Macdonald, C., Cappellato, L., Ferro, N. (eds.) Experimental IR Meets Multilinguality, Multimodality, and Interaction. 7th International Conference of the CLEF Initiative (CLEF 16). Springer, Berlin Heidelberg New York (Sep 2016)
20. Tellez, E.S., Miranda-Jiménez, S., Graff, M., Moctezuma, D., Suárez, R.R., Siordia, O.S.: A simple approach to multilingual polarity classification in twitter. Pattern Recognition Letters pp. – (2017), http://www.sciencedirect.com/science/article/pii/S0167865517301721
21. Tellez, E.S., Moctezuma, D., Miranda-Jímenez, S., Graff, M.: An automated text categorization framework based on hyperparameter optimization. arXiv preprint arXiv:1704.01975 (2017)
22. Ucelay, M.J.G., Villegas, M.P., Funez, D.G., Cagnina, L.C., Errecalde, M.L., Ramırez-de-la Rosa, G., Villatoro-Tello, E.: Profile-based approach for age and gender identification (2016)
23. op Vollenbroek, M.B., Carlotto, T., Kreutz, T., Medvedeva, M., Pool, C., Bjerva, J., Haagsma, H., Nissim, M.: Gronup: Groningen user profiling (2016)
24. Zahid, A., Sampath, A., Dey, A., Farnadi, G.: Cross-genre age and gender identification in social media (2016)