PRHLT at PR-SOCO: A Regression Model for Predicting Personality Traits from Source Code

Notebook for PR-SOCO at FIRE 2016

Maite Giménez
Pattern Recognition and Human Language Technology (PRHLT) Research Center
Universitat Politècnica de València
Camino de Vera s/n, 46022 Valencia, Spain
mgimenez@dsic.upv.es

Roberto Paredes
Pattern Recognition and Human Language Technology (PRHLT) Research Center
Universitat Politècnica de València
Camino de Vera s/n, 46022 Valencia, Spain
rparedes@dsic.upv.es

ABSTRACT

This paper describes our participation in the PAN@FIRE Personality Recognition in Source Code (PR-SOCO) 2016 shared task. We have proposed two different approaches to tackle this task, on the one hand, each code sample from each author was taken as an independent sample and it was vectorized using word n-grams; on the other hand, all the code from an author was taken as a unique sample, and it was vectorized using word n-grams together with hand-crafted features that may determine the personality traits of an author. Regardless of the approach, a regression model was trained to classify the personality traits of the author of a sample of source code. All the systems we have submitted to be evaluated have achieved a root mean square error (RMSE) below the mean RMSE of the participants of the shared task. Moreover, one of our runs, the one that included the hand-crafted features, held the best result in the personality trait *Agreeableness*. This suggests that in the absence of enough independent samples to train a machine learning system, hand-crafted features are able to obtain better results.

Keywords

PR-SOCO; Author profiling; Personality Recognition; Source Code; Natural Language Processing; Machine Learning; Regression

1. INTRODUCTION

One of the new emerging research areas in Natural Language Processing (NLP) is Personality Recognition (PR), which seeks to classify the personality traits of the author of a text. In psychology, Norman et al. (1963) [11] proposed a taxonomy for describing the personality along five dimensions known as "Big Five", which are: agreeableness, conscientiousness, extroversion, openness to experience, and emotional stability. Besides, this work determined that our personality traits have a strong influence on our individual behavior. The work carried out by Gill (2003) [8] outline that the personality is projected through the language. Therefore, by exploiting different kinds of NLP techniques, it is possible to infer the personality of the author of a text. In addition, Personality Recognition can be useful in various applications such as marketing, sociology, etc. [6, 7, 15, 18]. Also, PR can be inferred using texts extracted from different sources: social media, essays, blog posts, etc. [1, 2, 14]. Finally, it is noteworthy that previous studies [12] have already proven the impact of the personality traits in the behavior of developers in the FLOSS community [1].
Previously, there were some efforts to evaluate Personality Recognition systems in several shared tasks, using texts gathered from Twitter [17], YouTube Vlogs, and Mobile Phone interactions [4]. However, the Personality Recognition in Source Code (PR-SOCO) shared task was the first competition where the objective was to determine the personality of developers from the source code they wrote, laying groundwork for a fair comparison between different approaches and future work.
In this paper we describe our participation for addressing the PR-SOCO task. The rest of the paper is organized as follows. Next section is devoted to define the Personality Recognition task. In Section 4 the model proposed is described. Following, in Section 5, the results achieved are presented. Finally, in Section 6 our results are discussed, and future work is proposed.

2. TASK DEFINITION

The main objective proposed by the organizers of the PR-SOCO shared task was to predict the personality traits of developers given a collection of their source code. The personality of a developer was determined following the *Five Factor Theory or Big Five* [5, 11, 3] which is the most widely accepted in psychology. Therefore, five traits define the personality of an author. Those traits are: agreeableness (A), conscientiousness (C), extroversion (E), openness to experience (O), and emotional stability / neuroticism (N). Each trait was labeled within a range between 20 and 80. The models were evaluated by the organizers using two metrics: the average Root Mean Squared Error (RMSE) as well as the Pearson Product-Moment Correlation (PC). For further information about the task, please review the overview paper of the task [16].

3. DATA

The organizers have gathered 60 samples of source code from 60 different programmers. In order to train the partic-

[1]Free/Libre Open Source Software https://www.gnu.org/philosophy/floss-and-foss.en.html

ipants' models, 49 samples were provided, and 21 were held to validate the results. Each sample consists of a collection of source code written in Java. In Table 1 the total number of training and test samples is shown.

Table 1: Dataset distribution

Dataset	Source Code	Authors
Train	1,741	49
Test	751	21

We have studied the distribution of the number of samples available for each value of each trait to classify depending on whether we considered the number of code samples as independent (number of pieces of source code) or not (number of authors). Figures 1 and 2 show the number of samples available for the trait *Agreeableness*. Similarly, the rest of the traits presented an equivalent distribution of the number of training samples available. It should be noted that the number of authors, and therefore the number of training samples available might be insufficient to adjust the parameters of a machine learning system adequately. If we consider each sample of code as an independent training sample, we will have more training samples available, which might be useful for fighting *the curse of dimensionality*[9]. This has led us to two different approaches that will be described in Section 4.

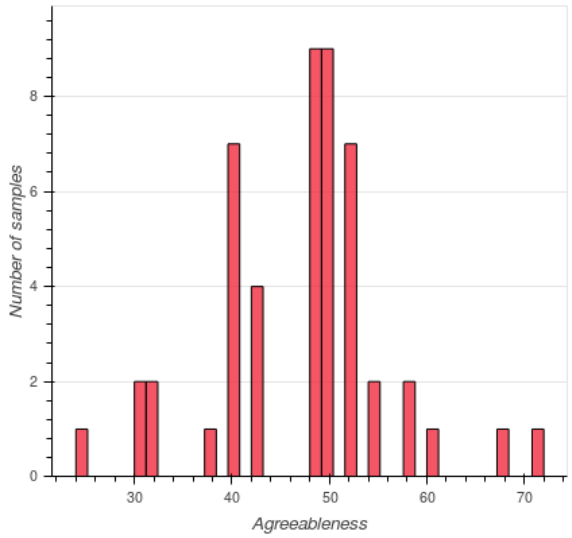

Figure 1: Num. of authors for each value of Agreeableness to classify (Author-Based approach).

Noteworthy, we are not exploiting any external dataset or resource to either train or fine-tune our models.

4. SYSTEM DESCRIPTION

Provided that the number of data samples available for training machine learning models is crucial, two approaches were evaluated. We have proposed an Author Based (AB) approach and a Code Based (CB) approach.

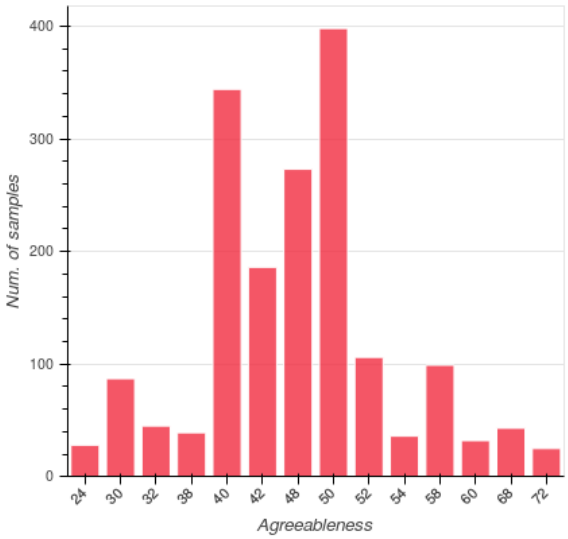

Figure 2: Num. of code samples for each value of Agreeableness to classify (Code-Based approach).

Author-Based approach uses all the samples of code from an author including hand-crafted features in addition to the words n-grams. The features considered were: the number of samples of code that implemented the same class (hf_1), the number of allocations (hf_2), the number of loops (hf_3), the appearance of pieces of code suspicious of plagiarism (hf_4)[2], the number of imports (hf_5), the number of functions (hf_6), the number of exceptions handled (hf_7), the number classes developed (hf_8), the number of different classes developed (hf_9), the number of comment lines (hf_{10}), and the number of prints (hf_{11}).

Code-Based approach assumed independence between the samples. This naïve assumption allowed us to train with 1,741 samples. The CB approach relies solely on the n-grams found in each piece of code, without considering any kind of aggregated information from each author. It generates a prediction for each sample of source code. Therefore, the final prediction for an author is the mean of all the predictions obtained for each piece of code that this author wrote.

As text representation, several vectorizer methods were evaluated for each approach. The vectorizers considered were: the term frequency-inverse document frequency (tf-idf) from one to four words (tfidf-words), the tf-idf from one to four n-grams of words ignoring the terms that have a frequency strictly higher than the threshold 0.5 and applying sub-linear scaling (sublinear-1:4), *idem* but exploring n-grams from one to six words (sublinear-1:6), the tf-idf from one to six characters (tfidf-chars), and a bag of words (BOW). We carried out a preprocessing phase where code snippets (e.g. a sequence of words that define a loop) were replaced by tokens. However, the systems that included this

[2]We supposed that those samples of code that instantiate classes that do not belong to the standard library are suspicious of plagiarism, e.g. the class *SeparateChainingHashTable*.

Table 2: RMSE achieved using a 5-fold validation over the train dataset following the Code Based approach. The mean RMSE and the standard deviation for the 5-fold validation for each trait is reported.

Model	Agreeableness	Conscientiousness	Extroversion	Neuroticism	Openness
sublinear-1:6 & ridge	6.10 (±0.67)	4.81 (±0.41)	5.55 (±0.89)	8.30 (±0.95)	4.93 (±0.52)
sublinear-1:4 & ridge	6.08 (±0.65)	4.82 (±0.44)	5.53 (±0.87)	8.26 (±0.94)	4.95 (±0.55)
sublinear-1:6 & LR.	6.11 (±0.85)	4.79 (±0.47)	5.94 (±0.89)	8.54 (±1.02)	4.85 (±0.43)
sublinear-1:4 & LR.	6.07 (±0.81)	4.83 (±0.47)	5.89 (±0.84)	8.49 (±1.01)	4.91 (±0.44)
sublinear-1:6 & RFR.	6.10 (±0.67)	5.00 (±0.72)	5.55 (±0.89)	8.30 (±0.95)	4.93 (±0.52)

phase obtained worse results that those systems without preprocessing. This phenomenon was previously reported in the author profiling literature [1, 10]. Our results confirm that the preprocessing phase also has a negative impact on the personality recognition task from source code.

Moreover, both approaches used a regression model to classify the authors automatically. The machine learning algorithms considered were: an Epsilon-Support Vector Regression (SVR) model, a Linear Regression (LR) model, a Linear Least Squares model with l2 regularization and $\alpha = 0.5$ (Ridge), Linear model trained with L1 prior as regularizer and $\alpha = 0.5$ (Lasso), a Multi-layer Perceptron classifier (MLP), a Decision Tree Regressor (DTR), and a Random Forest Regressor (RFR). The task was also evaluated as a classification problem using Support Vector Machines, and Random Forest. Nevertheless, the classification approach behaved worse than the regression approach. Therefore, this classification approach was discarded.

We have developed a pipeline using scikit-learn [13]. In the CB approach, we have selected the best combination of n-grams and the regression model using a 5-cross validation. The selection of the models was a compromise solution. We selected those models that achieved better global RMSE computed as the mean of the RMSE for each trait and for each fold:

$$\sum_{trait \in A,C,E,N,O} \frac{\sum_{fold=1}^{5}(RMSE_{trait\&fold})/5}{5}$$

This has allowed us to obtain models with a competitive performance for all traits measured with the RMSE. Our systems were only optimized for the RMSE, which might affect the performance using the Pearson Correlation since there is no reciprocity between the RMSE and the Pearson Correlation. Conversely, in the AB approach, the best hand-crafted combination was selected applying an ablation test, and these features were concatenated to the word n-grams of the best model obtained for the CB approach.

5. RESULTS

Hereafter, we will describe the results achieved by our best models. Table 2 shows the RMSE of our best models at development time. Due to the computational complexity of performing the grid search over two metrics, we have only used the RMSE to adjust our models.

After selecting the best model for the Code-Based approach, we have selected the hand-crafted features that improved the classification in the Author-Based approach. The hand-crafted features selected were: the number of samples of code that implemented the same class hf_1, the appearance of pieces of code suspicious of plagiarism hf_4, the number of classes developed hf_8, and the number of different classes developed hf_9.

We submitted five different models. Those that performed better during the development phase, which were:

1. **run 1**: a Code-Based approach using sublinear-1:4 and Ridge.
2. **run 2**: a Code-Based approach using sublinear-1:6 and Ridge.
3. **run 3**: an Author-Based approach using sublinear-1:4, the following hand-crafted features: $hf_1 \oplus hf_4 \oplus hf_8 \oplus hf_9 \oplus hf_{10}$ and Ridge.
4. **run 4**: a Code-Based approach using sublinear-1:4 and Logistic Regression.
5. **run 5**: a Code-Based approach using sublinear-1:6 and Logistic Regression.

Two baselines were provided by the organizers: a bag of words 3-grams with frequency weight (bow), and an approach that always predicts the mean value observed in the training data (mean). The evaluation results for each personality trait over the test set can be found in Table 3.

Eleven teams have presented their respective systems. In total, 48 systems were submitted for evaluation. All the systems we have submitted have performed better than the mean of the systems proposed using the RMSE.
Despite the results achieved during the development phase, our best performing system was the one that followed the Author-Based approach. This system was able to achieve the best RMSE result in the personality trait *Agreeableness*. Nevertheless, our systems' predictions did not find a correlation with the gold standard following the Pearson coefficient metric. Besides, neither the baselines proposed nor the best performing participants were able to find a significative correlation. The best correlation found by the participants was 0.62 for the trait *Openness*, which can not be considered a strong positive correlation.

6. DISCUSSION AND FUTURE WORK

In this paper we have presented our participation in the PAN@FIRE Personality Recognition in Source Code 2016 shared task. Two approaches were proposed an Author-Based approach and a Code-Based approach. The AB approach performed better for all the traits. This could be explained because the samples we used to train the systems that followed the Code-Based approach were not independent. Therefore, the results we obtained in the development

Table 3: Evaluation of our participation in the PR-SOCO shared task. The first five rows, run 1 up to run 5, show the results achieved by our systems. The traits are: agreeableness (A), conscientiousness (C), extroversion (E), neuroticism (N), and openness to experience (O). Moreover, the performance of the baseline systems are included, as well as the minimum, maximum and mean performance obtained by the participants at the shared task.

(a) RMSE achieved in the test dataset

Model	A	C	E	N	O
(CB) run 1	9.29	9.02	8.75	10.67	7.85
(CB) run 2	9.36	8.99	8.79	10.46	7.67
(AB) run 3	**8.79**	8.69	9.0	10.22	7.57
(CB) run 4	9.62	8.86	8.69	10.73	7.81
(CB) run 5	9.71	8.89	8.65	10.65	7.79
baseline bow	9.0	8.47	9.06	10.29	7.74
baseline mean	9.04	8.54	9.06	10.26	7.57
min	8.79	8.38	8.60	9.78	6.95
max	28.63	22.36	28.80	29.44	33.53
mean	9.72	10.74	12.27	12.75	10.49

(b) Pearson Correlation achieved in the test dataset.

Model	A	C	E	N	O
(CB) run 1	0.03	-0.23	0.31	-0.22	-0.12
(CB) run 2	0.0	-0.19	0.28	-0.07	0.05
(AB) run 3	0.33	-0.12	0.18	0.09	0.03
(CB) run 4	-0.03	-0.09	0.28	-0.15	-0.05
(CB) run 5	-0.06	-0.12	0.3	-0.16	-0.02
baseline bow	0.20	0.17	0.12	0.06	-0.17
baseline mean	0.0	0.0	0.0	0.0	0.0
min	-0.32	-0.31	-0.37	-0.29	-0.36
max	0.38	0.33	0.47	0.36	0.62
mean	-0.01	-0.01	0.06	0.04	0.09

phase correspond to over-fitted systems.
However, provided that we did not have enough samples we still need to include proper techniques for data augmentation. If we would be able to get more labeled data, new approaches could be studied such as deep learning methods and word embeddings for text representation.
Noteworthy, the minimum error achieved by the participants' proposals in the RMSE is close to the baseline models for all the personality traits, and only for some traits a correlation with the gold standard was found. This highlights the complexity of the task. Therefore, personality recognition in source codes is an open problem and new NLP approaches could improve the performance of the systems.

7. REFERENCES

[1] S. Argamon, S. Dhawle, M. Koppel, and J. W. Pennebaker. Lexical predictors of personality type. In *Proceedings of the 2005 Joint Annual Meeting of the Interface and the Classification Society of North America*, 2005.

[2] S. Argamon, M. Koppel, J. W. Pennebaker, and J. Schler. Mining the blogosphere: Age, gender and the varieties of self-expression. *First Monday*, 12(9), 2007.

[3] G. J. Boyle, G. Matthews, and D. H. Saklofske. *The SAGE Handbook of Personality Theory and Assessment: Personality Measurement and Testing*, volume 2. Sage, 2008.

[4] F. Celli, B. Lepri, J.-I. Biel, D. Gatica-Perez, G. Riccardi, and F. Pianesi. The workshop on computational personality recognition 2014. In *Proceedings of the 22nd ACM international conference on Multimedia*, pages 1245–1246. ACM, 2014.

[5] P. T. Costa and R. R. McCrae. The revised neo personality inventory (neo-pi-r). *The SAGE handbook of personality theory and assessment*, 2:179–198, 2008.

[6] S. Cruz, F. Q. da Silva, and L. F. Capretz. Forty years of research on personality in software engineering: A mapping study. *Computers in Human Behavior*, 46:94–113, 2015.

[7] R. Fuchs. Personality traits and their impact on graphical user interface design. In *2nd Workshop on Attitude, Personality and Emotions in User Adapted Interaction*, 2001.

[8] A. J. Gill. *Personality and language: The projection and perception of personality in computer-mediated communication*. PhD thesis, University of Edinburgh, 2003.

[9] E. Keogh and A. Mueen. Curse of dimensionality. In *Encyclopedia of Machine Learning*, pages 257–258. Springer, 2011.

[10] A. McEnery and M. Oakes. Authorship studies/textual statistics. 2000.

[11] W. T. Norman. Toward an adequate taxonomy of personality attributes: Replicated factor structure in peer nomination personality ratings. *The Journal of Abnormal and Social Psychology*, 66(6):574, 1963.

[12] O. H. Paruma-Pabón, F. A. González, J. Aponte, J. E. Camargo, and F. Restrepo-Calle. Finding relationships between socio-technical aspects and personality traits by mining developer e-mails. In *Proceedings of the 9th International Workshop on Cooperative and Human Aspects of Software Engineering*, pages 8–14. ACM, 2016.

[13] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[14] B. Plank and D. Hovy. Personality traits on twitter - or - how to get 1,500 personality tests in a week. In *Proceedings of the 6th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 92–98, 2015.

[15] D. Preotiuc-Pietro, J. Eichstaedt, G. Park, M. Sap, L. Smith, V. Tobolsky, H. A. Schwartz, and L. Ungar. The role of personality, age and gender in tweeting about mental illnesses. *NAACL HLT 2015*, page 21, 2015.

[16] F. Rangel, F. González, F. Restrepo, M. Montes, and P. Rosso. Pan at fire: Overview of the pr-soco track on personality recognition in source code. In *Working notes of FIRE 2016 - Forum for Information Retrieval Evaluation, Kolkata, India, December 7-10, 2016*, CEUR Workshop Proceedings. CEUR-WS.org, 2016.

[17] F. Rangel, P. Rosso, M. Potthast, B. Stein, and

W. Daelemans. Overview of the 3rd author profiling task at pan 2015. In *CLEF*, 2015.

[18] R. S. Rubin, D. C. Munz, and W. H. Bommer. Leading from within: The effects of emotion recognition and personality on transformational leadership behavior. *Academy of Management Journal*, 48(5):845–858, 2005.