UAEMex System for Identifying Personality Traits from Source Code

Eder Vázquez Vázquez[1], Omar González Brito[2], Jovani A. García[3], Miguel García Calderón[4], Gabriela Villada Ramírez[5], Alan J. Serrano León[6], René A. García-Hernández[7], Yulia Ledeneva[8]

Universidad Autónoma del Estado de México, UAPT Tianguistenco.

Instituto Literario, 100, Toluca, Edo. Méx. 50000, México.

eder2v@hotmail.com[1], gonzalezbritoomar@gmail.com[2], jovani_2807@hotmail.com[3], tonsquemike@outlook.com[4], inggaby.vr@gmail.com[5], alan.serrano.leon@outlook.com[6], renearnulfo@hotmail.com[7], yledeneva@yahoo.com[8]

ABSTRACT

This paper describes the UAEMex participation on Personality Recognition Source Code (PR-SOCO 2016) task, where the principal challenge is to identify the five personality traits using the source code of a developer. In the first phase of the task, a training dataset with 50 programs and the degree values of the personality incidence for each trait were provided. In the second phase, a test dataset with 21 programs must be classified. Our method consists in extracting only 41 features from the source code including the comments in order to classify it (we test 4 models). Using the evaluation metrics proposed by PR-SOCO, our system is ranked between the best systems for both evaluation metrics. Finally, using the RMSE and the PC metric we propose a ranking measure.

Keywords

PR-SOCO; Support Vector Machine; Symbolic Regression; KNN; Neural Networks; Personality Trait; Genetic Algorithms.

1. INTRODUCTION

Personality is an inherent aspect of human nature that has an influence on its activities. It means, personality is a set of characteristics that describes one person, and makes it different from others [1]. Nowadays, identifying the degree of personality traits for determining if a candidate fits with a job is such important as skills and experience [2]. After decades of research, the Big-Five Theory is the most accepted model for assessing the personality [2]. This model has a hierarchical organization of personality traits with five classes: Extroversion (E), Agreeableness (A), Conscientiousness (C), Neuroticism (N), and Openness to experience (O) [3].

Given a few set of java source codes in PR-SOCO task, the main objective is to identify the degree of presence of five classes of personality [4]. In order to get an approximation of what aspects determine the personality, the NEO-PI R test may be answered (this test is based on the Big-Five theory) to measure the personality traits [3]. There are many structured surveys based on NEO-PI R in several Web pages, available on-line for everybody to predict the personality of the user. Using these aspects, we propose to extract 41 features as the main information for training four classifiers.

In this paper, we present the working notes of the UAMEX participation on the PR-SOCO 2016 task.

This paper is organized as follows. In section 2, the methodology is described. In section 3, the results for the test dataset experiments are presented. In section 4, using the evaluation metrics proposed by PR-SOCO, we rank the results with others systems by personality traits. In section 5, the conclusions are presented.

2. METHODOLOGY

The proposed methodology is divided in four steps: Corpus Analysis, Feature Extraction, Feature Representation and Classification.

2.1 Corpus Analysis

The training dataset is composed of 1741 java source codes of 50 developers that where evaluated with the Big-Five Theory personality traits where each trait ranges between value of 20 and 80. However, the number of different values by personality trait in the samples is small, we decided to manage each program by separated, to get a good representation (See table 1). There are different numbers of values per class on every personality trait, the distribution is shown in table 1.

Table 1. Source code distribution for every personality trait.

Personality Trait	Number of different values
Neuroticism	13
Extroversion	14
Openness	11
Agreeableness	14
Conscientiousness	12

2.2 Feature Extraction

Using few source codes of our team members, we identify some personal features in order to identify some similar elements. As result, we detected the indentation, identifier and comment features are important to determine the author of such codes. These features can be extracted independently of the content or objective of the source code. The first 25 features were calculated using average and the last 16 were calculated using frequency. Extracted features can be classified as:

Indentation Features: space in code, space in the comments, space between classes, space between source code blocks, space between methods, space between control sentences and space in clustering characters “(), [], {}”. These features are measured with the average.

Identifier Features: The presence of underscore, uppercase and lowercase in the name of an identifier is measured in binary way. Also, we extract the average number of characters and the average length in the name of an identifier as features. These features are extracted for class, methods and variable names. Also, the percentage of number of initialized variables is extracted.

Comment Features: The presence of line and block comments are extracted as binary features. Also, the presence of comments with all letters in uppercase is extracted as binary feature. Finally, the average of size of the comments is extracted as feature.

2.3 Features Representation

For every source code, 41 features are extracted for representing in a vector space model, where the Source Code S_i is represented by one of the 41 features f_j [5].

2.4 Classification

Once the source codes are represented in a vector space model, we train the system with the next classifiers. The objective of test different classifiers is that if the extracted features are good features then we would get, in general, good results with these classifiers. It is worth to say that these classifiers have been widely used in other language processing tasks, especially we trust in the Symbolic Regression model since the training dataset only has some few values per trait.

2.4.1 Symbolic Regression (SR)

Finding the structure, coefficients and appropriate elements of a model at same time that try to solve problem, is a challenge for which no efficient mathematical method exists, therefore traditional mathematical techniques are not the best in empirical modeling problems due to their nonlinearity. Because, there is a need with an artificial expert which can create or define a model from available data of specific task without appeal problem understand [6]. Symbolic Regression is an artificial expert type that evolve models from available data observations [7] [8], whose main objective is to find a model which describes the relationship between dependent variable and independent variables as accurately as possible [9].

Because Symbolic Regression works directly with Genetic Programming is possible to evolve equations or mathematical functions in order to estimate the behavior of a dataset. The symbolic regression technique standout as a viable solution to the problem of this work because it does not assume an answer problem, but also discover it [10].

2.4.2 Support Vector Machine (SVM)

SVM maps a set of examples as a set of points in the same space trying to get optimal hyper-plane. Optimal hyper-plane is defined as hyperplane with maximal separation between two classes [11]. SVM make predictions based on which side of the gap they fall on [12]. In this work, we used SVM implementation LIB-SVM [13].

2.4.3 K Nearest Neighbor (KNN)

Is one of the simplest machine learning algorithms known as lazy classifier where classification function is only approximated locally. KNN is trained using vectors on feature space; each vector must have a class label. The training phase consists on store feature vectors and class labels of training dataset. In classification phase is necessarily to define constant k and send an unlabeled vector to KNN algorithm for calculate the minimal distance between stored classes and input vector [14]. We use Weka implementation for KNN algorithm [15].

2.4.4 Back Propagation Neural Network (BP-NN)

Neural networks are an elemental processor that recipe a vector as input data. The feature vector is send at input layer and then every neuron processes a $k - input$ with $k - weight$ and returns a $k - output$. Neural networks are used to approximate functions according to the input data [16].

When neural network implements back-propagation error, the output of neural network is compared with desired output to calculate neural network error and then correct weights of every neuron in hidden layer [17].

3. RUN RESULTS

In this section, the results submitted for the PR-SOCO test dataset are described.

Run 1: This run was generated using symbolic regression (SR) over the vector space model but we eliminate the source codes of five developers according to the next criterion: the person who has a high presence in all the personality traits, the person who has a lower presence in all the personality traits, the person who has an average presence in all the personality traits, the person who has more source codes and the person who has few source codes.

Run 2: Similar to run 1, this run was generated using (SR) but for each personality trait the developers (between 12 and 20) with average presence of such trait were eliminated.

Run 3: For this run, the whole training dataset was used with Back Propagation Neural Network.

Run 4: The whole training dataset with KNN with constant $k = 3$ was used.

Run 5: We use a genetic algorithm, but it is not described because we find a mistake.

Run 6: The whole training dataset was used for classify with a SVM.

Root Mean Square Error (RMSE) and Pearson Correlation (PC) metrics were used by PR-SOCO task as evaluation of the ranking results. A minimum RMSE is desired for a system. In change, in PC metrics a closer value to 1 or -1 is desired. In table 2, the RMSE scores of our runs are presented, with the best scores highlighted in bold. As is possible to see, the first and six runs get the best scores, where the SR and SVM classifiers were used, respectably.

Table 2. RMSE results of submitted runs for test dataset.

Run	**N**	**E**	**O**	**A**	**C**
1	11.54	11.08	**6.95**	**8.98**	**8.53**
2	11.10	12.23	9.72	9.94	9.86
3	**9.84**	12.69	7.34	9.56	11.36
4	10.67	**9.49**	8.14	8.97	8.82
6	10.86	9.85	7.57	9.42	**8.53**

In table 3, the results with Pearson Correlation metric is showed, with the best score highlighted in bold.

Table 3. PC results of submitted runs for test dataset.

Run	N	E	O	A	C
1	-0.29	**-0.14**	**0.45**	0.22	0.11
2	-0.14	-0.15	0.04	0.19	**-0.30**
3	**0.35**	-0.10	0.28	**0.33**	-0.01
4	0.04	-0.04	0.10	0.29	-0.07
6	0.13	0	0	0	0

4. RANKING RESULTS

In PR-SOCO 2016, eleven teams participated in this task with two baseline: the baseline bow (*bl bow*) based on trigram of chars and the baseline mean (*bl mean*) based on a method that predicts the mean value of the observed values. In table 4, the best RMSE results of those teams for every personality trait are showed according to the rank. In general, our results (*uaemex*) were ranked in good positions outperforming the baseline, except for **E**xtroversion, in the case of **N**euroticism and **A**greeableness we were ranked in second position, in the case of **O**penness we get the first rank and for **C**onscientiousness we get the fourth position between two baselines.

Table 4. Best runs with RMSE metric.

Rank	N	E	O	A	C
1	9.78	8.6	**6.95** **uaemex**	8.79	8.38
2	**9.84** **uaemex**		7.16	**8.97** **uaemex**	8.39
3	9.97	8.69	7.19	9 *bl bow*	8.47 *bl bow*
4	10.04	8.8	7.27	9.04 *bl mean*	**8.53** **uaemex**
5	10.24	8.96	7.42	9.16	8.54 *bl mean*
6	10.26 *bl mean*	9.01	7.57 *bl mean*	9.32	8.59
7	10.27	9.06 *bl bow* *bl mean*		9.36	8.61
8	10.28		7.74 *bl bow*	9.39	8.69
9	10.29 *bl bow*	9.22	8.19	9.55	8.77
10	10.37	**9.49** **uaemex**	8.21	10.31	8.85
11	10.53	11.18	8.43	11.5	9.99
12	17.55	16.67	15.97	21.1	15.53
13	24.16	27.39	22.57	28.63	22.36

In table 5, the best PC results of those teams for every personality trait are showed according to the positive correlation results. In general, our results (*uaemex*) were ranked in good positions outperforming the baseline configurations. In the case of **N**euroticism, **O**penness, **A**greeableness and **C**onscientiousness we were ranked in second position except for the **E**xtroversion trait. In general, it is possible to observe that the rank of our results for the RMSE metric correspond with the rank of our results for the PC metric.

In PR-SOCO 2016, two evaluation metrics were used given two ways of ranking the results, the RMSE for measuring the average error between the observed and predicted values and the PC for measuring the correlation between variables. In this paper, we propose ranking the results using both RMSE and PC measures as:

$$Ranking = ((1 - PC) * RMSE)$$

This measure only is applied for positive correlation results in PC metric. Since RMSE is not normalized we propose to multiply both results. This ranking is a metric where best values are those closer to cero. Table 6 shows the best results evaluating with our proposing measure.

Table 5. Best runs with PC metric.

Rank	N	E	O	A	C
1	0.36	0.47	0.62	0.38	0.33
2	**0.35** **uaemex**	0.38	**0.45** **uaemex**	**0.33** **uaemex**	**0.32** **uaemex**
3	0.31	0.35	0.37	0.29	0.31
4	0.29	0.31	0.33	0.21	
5	0.27	0.31	0.3	0.21	0.21
6	0.23	0.16	0.29	0.19	0.19
7	0.14	0.12 *bl bow*	0.27	0.06	0.16
8	0.1	0.11	0.12	0	0.13
9	0.1	0.1	0.05	-0.05	0.07
10	0.09	0.08	0 *bl mean*	-0.07	-0.12 *bl mean*
11	0.06 *bl bow*	0.11	-0.15	0.08 *bl mean*	
12	0.05	**0** **uaemex**	-0.17 *bl bow*	-0.19 *bl bow*	-0.2 *bl bow*
13	0 *bl mean*	0 *bl mean*	-0.31	-0.28	-0.23

Table 6. Results with our proposal evaluation metric.

Ranking	N	E	O	A	C
1	**6.39** **uaemex**	5.32	**3.82** **uaemex**	5.88	6.24
2	6.54	5.59	4.60	**6.36** **uaemex**	6.78
3	6.74	6.03	4.79	6.71	7.03 *bl bow*
4	7.67	6.07	5.13	6.98	7.47
5	8.84	7.52	5.26	7.2 *bl bow*	7.55
6	8.91	7.97 *bl bow*	7.28	8.24	**7.59** **uaemex**
7	9.3	8.49	7.57 *bl mean*	8.49	8.54 *bl mean*
8	9.67 *bl bow*	9.06 *bl mean*	8.23	9.04 *bl mean*	11.33
9	9.74	9.32	8.43	9.26	-

10	9.93	**9.85** **uaemex**	*8.97*	*22.61*	-
11	10.26 *bl mean*	23.20	16.47	-	-
12	12.46	24.65	-	-	-
13	21.74	-	-	-	-

As we can see in table 6, our results get a better balance between RMSE and PC. In table 6, ***uaemex*** team is ranking in first position for **N**euroticism and **O**penness trait, in second place for **A**greeableness and sixth place for **C**onscientiousness. However, in this new ranking the **E**xtroversion do not outperform both baselines.

5. CONCLUSIONS

This paper presents the results in personality trait prediction. We describe the participation of the UAEMex at PR-SOCO 2016.

We know that submitted runs overcome the baseline despite that corpus has noise like repeated source code, obfuscated source code and it have little samples.

The training set has different classes of personality. There are unbalanced classes and there has not enough examples for class values. In this approach, we do not make preprocessing because it was considered that all information in corpus are relevant by the task. Personality Trait Prediction in source code is a new task and there are not reference approaches about this. It was difficult to identify what features would be extracted.

The best results in our runs obtained with the symbolic regression model because the training phase try to approximate the output of input vector.

Also, we propose a new ranking measure for combine a RMSE and PC measure in order to get an approximation for evaluation results. According to our experiments in train dataset, we note that it is better than RMSE or PC evaluating alone. RMSE is a minimization metric and PC is a maximization metric.

6. ACKNOWLEDGMENTS

Thanks to Autonomous University of the State of Mexico (UAEMex), Consejo Nacional de Ciencia y Tecnología (CONACyT) and Consejo Mexiquense de Ciencia y Tecnología (COMECyT) for support granted for this work.

7. REFERENCES

[1] Montaño, M., Palacios, J., Gantiva, C. 2009. Teorías de la personalidad. Un análisis histórico del concepto y su medición. Psychologia Avances de la disciplina, 81-107.

[2] Paul, C., R., M.R. 2008. NEO PI-R Revised Neo Personality Inventory. TEA Ediciones S.A.

[3] Hussain, S., Abbas, M., Shahzad, K., Syeda, A. 2012. Personality and career choices. African Journal of Business Management (AJBM) 6, 2255-2260.

[4] Rangel, F., González, F., Restrepo, F., Montes, M. and Rosso, P. 2016. 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.

[5] Salton, G., Wong, A., Yang, C.S. 1975. A vector space model for automatic indexing. Commun. ACM 18, 613-620.

[6] Dabhi, V.K., Vij, S.K. 2011. Empirical modeling using symbolic regression via postfix Genetic Programming. Image Information Processing (ICIIP), 2011 International Conference on, 1-6.

[7] Koza, J.R. 1992. Genetic programming: on the programming of computers by means of natural selection. MIT Press.

[8] Murari, A., Peluso, E., Gelfusa, M., Lupelli, I., Lungaroni, M., Gaudio, P. 2015. Symbolic regression via genetic programming for data driven derivation of confinement scaling laws without any assumption on their mathematical form. Plasma Physics and Controlled Fusion 57.

[9] Kommenda, M., Affenzeller, M., Burlacu, B., Kronberger, G., Winkler, S. M. 2014. Genetic programming with data migration for symbolic regression. In: Proceedings of the 2014 conference companion on Genetic and evolutionary computation companion, 1361-1366.

[10] Can, B., Heavey, C. 2011. Comparison of experimental designs for simulation-based symbolic regression of manufacturing systems. Computers and Industrial Engineering 61, 447-462.

[11] Hearst, M.A. 1998. Support Vector Machines. IEEE Intelligent Systems 13, 18-28.

[12] Cortes, C., Vapnik, V. 1995. Support-Vector Networks. Machine Learning 20, 273-297.

[13] Chang, C.-C., Lin, C.-J. 1977. LIBSVM: A library for support vector machines. ACM Trans. Intell. Syst. Technol. 2, 1-27.

[14] Stone, C.J. 1977. Consistent Nonparametric Regression. 595-620.

[15] Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I. H. 2009. The WEKA data mining software: an update. SIGKDD Explor. Newsl. 11, 10-18.

[16] McCulloch, W.S., Pitts, W. 1988. A logical calculus of the ideas immanent in nervous activity. In: James, A.A., Edward, R. (eds.) Neurocomputing: foundations of research, 15-27.

[17] Rumelhart, D.E., Hinton, G.E., Williams, R. J. 1986. Learning internal representations by error propagation. In: David, E.R., James, L.M., Group, C.P.R. (eds.) Parallel distributed processing: explorations in the microstructure of cognition, vol. 1, 318-362.