A Supervised Approach for Personality Recognition in Source Code using Code Analysis Tool at FIRE 2016

Rehana Delair
SNPIT&RC
Bardoli, Gujarat
+91 9904039419
rehanad10@gmail.com

Rutal Mahajan
SNPIT&RC
Bardoli, Gujarat
+91 9426393096
rutal.mahajan@gmail.com

ABSTRACT
Personality Recognition from the author's source code is a task organized by PR-SOCO team in conjunction with the FIRE 2016 Forum for Information Retrieval Evaluation. The aim is to identify author's personality traits from source code collection of a programmer. We have used various supervised learning approaches to train the regression model with different set of features extracted using static code analysis tool checkstyle. Based on these features, the trained regression model is used to predict the score for different personality traits. All the systems are evaluated using two evaluation metrics: Root Mean Squared Error (RMSE) and Pearson Product-Moment Correlation (PC). Our system has scored 0.62 and 0.33 PC in two personality traits, Openness and Conscientiousness respectively using M5Rules algorithm as regression model, which is the best score among all the submitted runs of our system as well as among all the participated systems.

Keywords
Personality Recognition, Machine Learning, Regression, Big-Five Personality traits

1. INTRODUCTION
There is a lot of work going on in the area of Personality Recognition [1] [5] [6] [7]. Personality traits influence most of the human activities such as the way people write [1], interact with each other, and the way they make a decision. The programmer's personality will affect the type of software project they chose to participate [6] or the way they write or structures their code.

There are many projects that use written text to identify author's personality. In "whose thumb is it anyway?" [5] personal weblogs are analyzed to predict personality traits. They have used the Support Vector Machine algorithm to predict personality traits. Main features are word based bi- and tri- grams. In "Finding relationships between socio-technical aspects and personality traits by mining developer e-mails." [6] they have used developer's emails to identify their personality.

Personality Recognition from the source code is different than other projects because the source code has limited scope. The Programmer doesn't have the choice to select their own word. They have to follow some of the pre-defined rules. Identifying Personality from the source code is a difficult task.

Personality can be defined along five traits using the Big Five Theory [3], which is the most widely accepted in psychology. The five traits are extroversion (E), emotional stability / neuroticism (S), agreeableness (A), conscientiousness (C), and openness to experience (O).

In order to collect different features from the given source code, checkstyle [2] is used. It is a code analysis tool which performs different checks on the source code. We have used weka [4] tool to train the regression model.

The rest of this paper is structured as follows: Section 2 outlines our approach on the Personality Recognition in Source Code. Section 3 presents tools used. Section 4 describes training and test data. Section 5 describes experiments and Section 6 describes official results of this task. Finally, we conclude in Section 7.

2. Approach

2.1 Overview
Main Process of Personality Recognition includes the following steps, which is shown in Figure 1:

1. **Collect individual corpora**
   In this step, we need to collect training data. In this case, we need source codes of different programmers which is training data provided by PR-SOCO committee [8].
2. **Collect associated personality ratings for each participant**
   This is the step where we collect personality ratings for each programmer. We have used Big-Five personality traits [3] to describe the personality of an individual. This data is also provided by PR-SOCO committee [8].
3. **Pre-processing**
   In this step, given file/data is converted into the efficient format for checkstyle [2]. It removes any separating lines from the source code and converts data into an actual JAVA file. We have also implemented a function to isolate one single program from the given training files of source code.
4. **Extract relevant features from the texts**
   In this step main features are identified from the given source code. We need to find different features of good source code which reflects authors' personality. For this purpose we have used a code analysis tool checkstyle [2]. It performs different checks on the source code such as how well the code is commented, how it is indented, naming conventions, etc. From this we have collected measures for different features.
5. **Build statistical models of the personality ratings based on the features**
   We have used different regression models to predict the personality traits like Support Vector Machine Regression, Gaussian Processes,M5 algorithm, M5' Rules and Random Tree. We have used JAVA API for Weka [4] to train different regression models.
6. **Test the learned models on unseen individuals**
   Using different features and trained regression model we predicted the score for different personality traits.

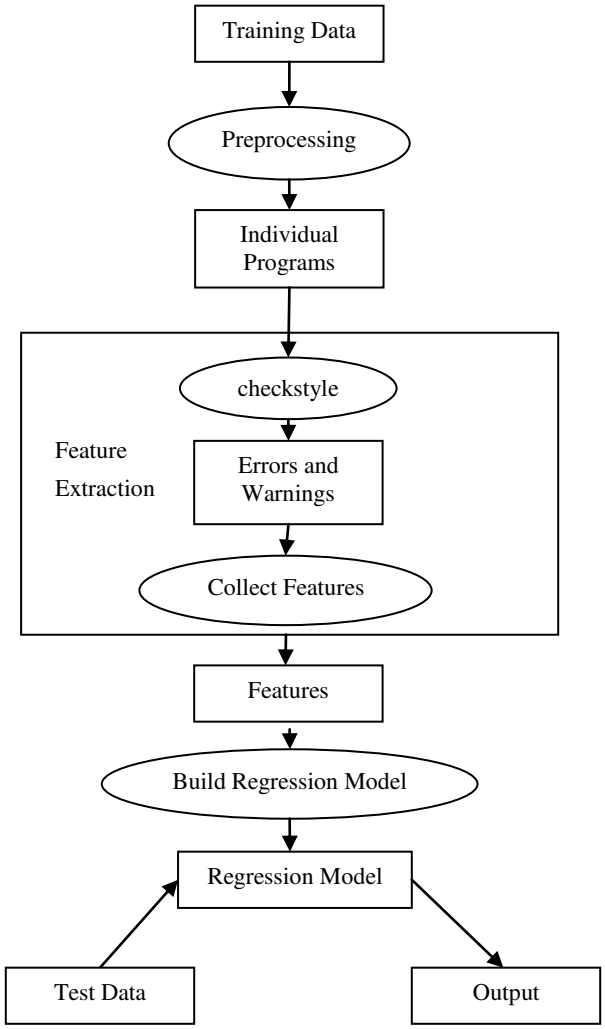

Figure 1. Flow Diagram of the Process

2.2 Features

We have used total 154 features of source code, which is extracted using static code analysis tool Checkstyle [2]. These features are categories into two categories to train the regression model: Style based features and Content based features. These are shown in the Table 1.

1. Style based Features
   It is the category of different features related to the style of the code. Such features are used to perform checks on code layout and formatting problems. It contains Indentation, Headers, Javadoc comments, white spaces, Block checks, etc.
2. Content based Features
   It is the category of different features related to the content of the source code. It performs checks on class design problems, method design problem, Annotations, Coding, Imports, Metrics, Modifiers, Naming conventions, Size violations and other miscellaneous features.

Table 1. Different Features Extracted by Checkstyle tool

Category	Feature Name	Number of features
Style based feature	Headers	2
	Javadoc comments	12
	White spaces	16
	Block Checks	6
	class design problems	9
Category based feature	Annotations	7
	Coding	43
	Imports	8
	Metrics	6
	Modifiers	2
	Naming conventions	15
	Size violations	8
	Miscellaneous	15
	Regular expression	5
Total		154

Single program is separated from the collection of source code and it is checked using checkstyle [2]. Errors and Warnings are counted and converted in per line of code format.

3. Data set

The training data set was provided by PR-SOCO committee itself that consists of source codes written in Java. The data consist of 49 documents that consist of a collection of source code of different authors. These source codes are labeled with personality traits of the programmer in a continuous range from 20 to 80.

Test data were also provided by PR-SOCO committee [8]. It is consists of 21 documents of a source code collection. We have used this data to evaluate our system.

4. Experiments

We have a collection of source code written by 49 different programmers along with their personality traits. We have used this data to train our model and then tested it on 21 unseen source codes. Two metrics were used to evaluate the system: the average Root Mean Squared Error (RMSE) as well as the Pearson Product-Moment Correlation (PC) between our software scores and the ground-truth scores. We have tested our system on a given test data. Results are discussed in the next section.

RMSE is the square root of the mean/average of the square of all of the error and PC is defined as a measure of the strength of a linear association between two variables.

We have used different Supervised Regression model to predict personality traits of different authors. These are Support Vector Machine, Gaussian Processes, M5P algorithm, M5Rule and Random tree algorithm. Support Vector Machine plots all the data items as a point in n-dimensional space. We have used default kernel settings in Support Vector Machine. M5P algorithm is decision tree based algorithm and M5Rule is rule based algorithm.

5. Results

We have submitted total five runs. This all runs use different regression algorithms. We have used Support Vector Machine, Gaussian Processes, M5P algorithm, M5Rule and Random tree algorithm for regression.

Results obtained for different runs of our system are shown in the Table 2. Two metrics are shown for each personality trait: RMSE/PC. It shows Root Mean Squared Error / Pearson Product-Moment Correlation values. At the bottom of the Table 2, measures for baselines: (a) a bag of character 3-grams with frequency weight; (2) an approach that always predicts the mean value observed in the training data are shown [8]. Our system has scored 0.62 and 0.33 PC in two personality traits, Openness and Conscientiousness respectively using M5Rules algorithm as regression model, which is the best score among all the submitted runs of our system as well as among all the participated systems.

In Neuroticism personality trait, our predicted scores are positively correlated with the ground truth scores. It gives nearly worst RMSE in Gaussian processes and SMO. In Extroversion personality traits, all regression models give different scores and it is weakly correlated with ground truth scores. Openness personality trait is strongly correlated with the ground truth score and gives good results. Agreeableness is negatively related with ground truth scores and it also gives worst RMSE. In Conscientiousness, predicted scores are positively correlated with ground truth scores.

Table 2. Official results of different runs of our system

Run	N	E	O	A	C
M5Rules	19.07/ 0.2	25.22/ 0.08	23.62/ **0.62**	21.47/ -0.15	22.05/ **0.33**
GP	26.36/ 0.19	16.67/ -0.02	15.97/ 0.19	23.11/ -0.13	21.72/ 0.1
M5P	18.75/ 0.2	25.22/ 0.08	20.28/ 0.54	21.47/ -0.15	22.05/ **0.33**
Random Tree	17.55/ 0.29	20.34/ -0.26	16.74/ 0.27	21.1/ -0.06	20.9/ 0.14
SMO	26.72/ 0.18	23.41/ -0.11	16.25/ 0.13	27.78/ -0.19	15.53/ 0.27
Baseline bow	10.29/ 0.06	9.06/ 0.12	7.74/ -0.17	9.00/ 0.20	8.47/ 0.17
Baseline mean	10.26/ 0.00	9.06/ 0.00	7.57/ 0.00	9.04/ 0.00	8.54/ 0.00
Best Results	9.78/ 0.36	8.60/ 0.47	6.95/ 0.62	8.79/ 0.38	8.38/ 0.33
Worst Results	29.44/ -0.29	28.80/ -0.37	33.53/ -0.36	28.63/ -0.32	22.36/ -0.31

6. Conclusion

Various supervised learning algorithms proved to be very capable of predicting personality traits scores for different authors from their given source code. Currently in our system we have not refined the effect of individual extracted features on different personality trait. Such refinement may yield better prediction results than the current submitted runs.

7. REFERENCES

[1] Celli F., Lepri B., Biel J. I., Gatica-Perez D., Riccardi G., Pianesi F. (2014). The workshop on computational personalty recognition 2014. Proc. Of the ACM Int. Conf. on Multimedia. Pp. 1245-1246.

[2] CheckStyle project , http://checkstyle.sourceforge.net/

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

[4] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, Ian H. Witten (2009); The WEKA Data Mining Software: An Update; SIGKDD Explorations, Volume 11, Issue 1

[5] Oberlander J., Nowson S. (2006). Whose thumb is it anyway? Classifying author personality from weblog text. Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions, pages 627–634, Sydney, July 2006. ©2006 Association for Computational Linguistics

[6] Paruma-Pab´on O.H., Gonz´alez F.A., Aponte J., Camargo J.E., Restrepo-Calle F. (2016). Finding relationships between socio-technical aspects and personality traits by mining developer e-mails. Workshop on Cooperative and Human Aspects of Software Engineering (CHASE), ICSE.

[7] Rangel F., Celli F., Rosso M., Potthast M., Stein B., Daelemans W. (2015). Overview of the 3rd Author Profiling Task at PAN 2015. CLEF 2015 Labs and Workshops, Notebook Papers. CEUR Workshop Proceedings. CEUR-WS.org, vol. 1391.

[8] Francisco Rangel, Fabio González, Felipe Restrepo, Manuel Montes and Paolo Rosso. PAN at FIRE: Overview of the PR-SOCO Track on Personality Recognition in SOurce Code. Working notes of FIRE 2016 - Forum for Information Retrieval Evaluation, Kolkata, India, December 7-10, 2016. CEUR Workshop Proceedings. CEUR-WS.org. 2016