Personality Recognition in Source Code Working Note: Team BESUMich

Shanta Phani
Information Technology
IIEST, Shibpur
Howrah 711103, West Bengal, India
shantaphani@gmail.com

Shibamouli Lahiri
Computer Science and Engineering
University of Michigan
Ann Arbor, MI 48109
lahiri@umich.edu

Arindam Biswas
Information Technology
IIEST, Shibpur
Howrah 711103, West Bengal, India
abiswas@it.becs.ac.in

ABSTRACT

In this paper, we describe the results of source code personality identification from Team BESUMich. We used a set of simple, robust, scalable, and language-independent features on the PR-SOCO dataset. Using leave-one-coder-out strategy, we obtained minimum RMSE on the test data for extroversion, and competitive results for other personality traits.

CCS Concepts

•**Computing methodologies** → **Natural language processing;** *Supervised learning by regression;*

Keywords

personality; source code; regression; RMSE; Pearson correlation; extroversion; neuroticism; openness; agreeableness; conscientiousness

1. INTRODUCTION

Personality is an important element of human sociology and psychology. It determines and underscores our day-to-day decisions, shopping and dating behaviors, educational aptitude, and emotional intelligence – to name a few. It is therefore no coincidence that the source code a programmer writes tends to be influenced by his/her personality. While the traditional Author Profiling task consists of predicting an author's demographics (e.g., age, gender, personality) from his/her writing, in the PR-SOCO shared task [15] the goal was to predict a programmer's personality from his/her source code. Personality traits influence most human activities, including but not limited to the way people write [4, 14], interact with others, and make decisions. For example in the case of programmers, personality traits may influence the criteria they use to select which open-source software projects to participate [11], and the way they write and organize their code.

In PR-SOCO, given a source code collection of a programmer, the goal was to identify his/her personality. Personality was defined according to five traits using the Big Five Theory [6]: extroversion (E), neuroticism (S), agreeableness (A), conscientiousness (C), and openness to experience (O). Each programmer was rated on a numeric scale on each of the five traits. Training and test data consisted of such ratings, along with code snippets from the developers. Since the response variable was a real number rather than a class label, we used a regression framework to model the supervised learning problem. We used a set of simple, robust, scalable, and language-independent features (Section 3), and optimized the root mean squared error (RMSE) averaged across all five traits in a leave-one-out cross-validation strategy. While applied on the test data, one of our runs achieved the minimum RMSE for extroversion.

The rest of this paper is organized as follows. We discuss relevant literature in Section 2. Section 3 gives details on the PR-SOCO task, especially the data and task description. We also describe our features, regressors, and experimental methodology in this section, especially delineating why we chose these features instead of code-style features. Section 4 provides experimental evaluation, and important insights that we gained along the way. We conclude in Section 5, outlining our contributions, limitations, and directions for future research. Relevant terminology is introduced as and when they first appear in the paper.

2. RELATED WORK

Personality recognition usually falls under the purview of *author profiling* [2, 3, 8, 14, 16]. Argamon et al. [2] showed that authors of informal texts could be successfully classified according to high or low neuroticism, and high or low extroversion. Four different sets of lexical features were used: a standard function word list, conjunctive phrases, modality indicators, and appraisal adjectives and modifiers. Appraisal use was found to be the best predictor for neuroticism, and function words worked best for extroversion. An SVM SMO classifier was used on essays written by college students.

Argamon et al. [3] extended this study in 2009 to take into account gender, age, native language, and personality. Three different corpora were used, in conjunction with content-based and style-based features. Bayesian Multinomial Regression (BMR) was used as classifier [9]. Style features were found to be very informative for personality traits. Most discriminative style features indicated that neurotics tended to refer to themselves.

Estival et al. [8] created an email dataset consisting of ten traits – five demographic (gender, age, geographic origin, level of education, native language), and five psychometric (the same ones mentioned in Section 1). They further designed a Text Attribution Tool (TAT), and subjected their data to this tool for rigorous validation, normalization, linguistic analysis, processing, and parsing. Three types of features – character-level, lexical, and structural – were extracted. It was shown that a combination of features performed best, and beat the baseline.

Rangel et al. [16] presented the Author Profiling Task at PAN 2013. The task consisted of age and gender classification in English and Spanish, and a special exercise on identifying adult-adult sexual conversations, and fake profiles for sexual predators. The task was extended by Rangel et al. in 2015 [14] to include four languages (English, Spanish, Italian, and Dutch), Big Five Personality traits, and Twitter users. The participants used content-based

Table 1: Statistics of the distribution of the number of code snippets in the PR-SOCO dataset. α represents the *power-law exponent* of the distribution. We also give the corresponding p-value (> 0.05 indicates significance).

Training data							
Min	Median	Mean	Max	SD	TOTAL	α	p-value
5	29	35.53	121	24.35	1741	2.86	0.91
Test data							
Min	Median	Mean	Max	SD	TOTAL	α	p-value
13	28	35.76	108	22.98	751	3.06	1.0

features (bag of words, word n-grams, term vectors, *tfidf* n-grams, named entities, dictionary words, slang words, ironic words, sentiment words, emotional words), and style-based features (frequencies, punctuation, POS, verbosity measures, several different tweet-specific statistics such as mentions, hashtags, and URLs). The highest accuracies in gender identification were achieved in Dutch and Spanish with values over 95%.

While all the above studies are important, and ground-breaking in some cases, we found none that looked into personality recognition from *source code*. From that perspective, the PR-SOCO shared task breaks a unique ground [15].

3. TASK DESCRIPTION

The PR-SOCO task [15] released a set of text files for 70 programmers – 49 as training data, and 21 as test. Each text file consisted of several source code snippets. The number of code snippets vary significantly from programmer to programmer. We show the distribution of snippets in Table 1. It is to be noted that the distribution forms a *power law* with exponent $\alpha = 2.86$ for the training data, and 3.06 for the test data (statistically significant in both cases; cf. [5]). Furthermore, there is considerable similarity among the programmers in the way they wrote code. This stems from two factors: (a) the programmers were given standardized coding questions (*prompts*) to implement, and (b) they were not precluded from using the Internet and copy-pasting code thereof. This resulted in substantial similarity between programmers. Moreover, oftentimes programmers wrote comments and named variables in non-English languages (we detected Spanish in manual investigation), and also submitted run information (which should ideally remain separate from the code).

All the above observations indicate that the data contains much noise. While we could have opted for a serious filtering and preprocessing step, such procedure was considered potentially harmful, because we could end up removing useful information such as coding style and unique developer signature. Note also that much of the source code is not natural language, so standard NLP tools such as parsers, named entity recognizers, and POS taggers would have been useless in such a scenario. Explicit code style indicators such as commenting and indentation patterns could have been useful, but the possibility of copy-pasting code from the Internet renders such features useless. Since comments and run information were intermixed with code, we needed a set of simple, robust, powerful, scalable, and language-independent features.

We are of the opinion that the only type of features that can offer all five of the above desiderata comes from word and character n-grams. They kill two birds with one stone: they are robust and resistant against copy-pasting from the Internet (because of the *shingling* property much used in plagiarism research [1]), and they are very effective at discriminating between author styles (as evidenced in authorship attribution studies [7, 13, 17]).

We therefore experimented with the two following categories of features: (1) **Bag of words**, and (2) **Character n-grams** (n = 1, 2, 3) with and without space characters and punctuation symbols. For each category, we experimented with lowercase and original case formatting, and three representations: binary (presence/absence), term frequency (tf), and tfidf. **Word n-grams** (n = 2, 3), and combination of different types of features (*feature fusion*; cf. [10]) could not be explored due to sparsity and runtime issues, which we would like to investigate in future.

We used three different regression models (*general linear models*) from the scikit-learn package [12]: **Linear Regression**, **Ridge Regression**, and **Lasso**. For Linear and Ridge Regression, we used default parameter settings. For Lasso, we tuned the α parameter as described in the next section. In the next section, we will see how the combinations of different features and regressors perform.

4. RESULTS

As mentioned in Section 1, we performed leave-one-coder-out cross-validation on the training data to find out the optimal feature-regressor combination, as well as the optimal parameter settings. We used the average across five RMSEs (for five personality traits) as our objective function. The reason we did not use Pearson Correlation Coefficient (ρ) or its square (R^2) is because there exists some debate as to whether we should use pure R^2 or adjusted R^2. RMSE avoids this debate. We would like to *minimize* the mean RMSE.

The main results are shown in Table 2 through Table 4. Note that overall, Linear Regression performs the worst, with high RMSEs across most feature combinations. This is expected, because the output space should be highly non-linear in terms of features. Ridge Regression and Lasso perform much better, with the best values coming out of Lasso using character unigrams (lowercased) – for binary, tf, and tfidf. This is a rather surprising finding, as it shows two things: (a) a handful of very simple character unigrams can capture very complex and highly non-linear output spaces, and (b) character unigrams beat more complex features in expressive power.

As next step, we proceeded to tune the Lasso parameter α that governs the *shrinkage* of coefficients. Note from Table 2 to Table 4 that the lowest RMSE came from lowercased character unigrams and *tfidf*. Hence, we used this combination, and tweaked the α parameter of Lasso. We obtained the following five top-performing combinations:

1. all characters, Lasso $\alpha = 0.05$, mean RMSE = 8.38.

2. all non-space characters, Lasso $\alpha = 0.05$, mean RMSE = 8.38.

3. all characters, Lasso $\alpha = 0.1$, mean RMSE = 8.4.

4. all non-space characters, Lasso $\alpha = 0.1$, mean RMSE = 8.4.

5. all characters, Lasso $\alpha = 0.01$, mean RMSE = 8.41.

Table 2: RMSE of leave-one-out cross-validation on the training data (default parameter settings for Lasso) with **binary** feature representation. Minimum values have been boldfaced. AW = all words, AC = all characters, SS = all characters except space, PP = all characters except punctuation, SP = all characters except space and punctuation.

Feature Representation	Feature Category	Feature Type	Linear Regression	Ridge Regression	Lasso
Binary (Presence/Absence)	Word unigrams	AW	2.82e12	8.77	8.89
	Word unigrams (lowercased)	AW	5.53e12	8.78	8.95
	Character unigrams	AC	5.83e12	8.8	8.66
		SS	1.34e13	8.8	8.66
		PP	4.14e12	8.82	8.65
		SP	1.21e12	8.82	8.65
	Character bigrams	AC	5.48e11	8.97	8.64
		SS	5.16e11	9.4	8.89
		PP	3.51e11	9.75	8.64
		SP	4.19e11	9.39	8.86
	Character trigrams	AC	3.91e12	8.65	8.72
		SS	4.72e12	8.61	8.68
		PP	5.32e12	8.6	8.83
		SP	7.53e12	8.73	8.81
	Character unigrams (lowercased)	AC	7.99e12	8.73	8.54
		SS	1.54e13	8.73	8.54
		PP	2.43e13	8.66	**8.51**
		SP	1.02e13	8.66	**8.51**
	Character bigrams (lowercased)	AC	2.50e11	9.11	**8.51**
		SS	2.80e11	10.11	8.82
		PP	2.12e11	9.82	8.6
		SP	4.85e11	9.89	8.87
	Character trigrams (lowercased)	AC	7.00e12	8.72	8.73
		SS	6.35e12	8.69	8.77
		PP	5.17e12	8.7	8.81
		SP	5.55e12	8.86	8.92

Table 3: RMSE of leave-one-out cross-validation on the training data (default parameter settings for Lasso) with **tf** feature representation. Minimum value has been boldfaced. AW = all words, AC = all characters, SS = all characters except space, PP = all characters except punctuation, SP = all characters except space and punctuation.

Feature Representation	Feature Category	Feature Type	Linear Regression	Ridge Regression	Lasso
Tf	Word unigrams	AW	2.63e11	9.07	8.96
	Word unigrams (lowercased)	AW	4.50e11	9.14	9.01
	Character unigrams	AC	8.75	8.73	8.6
		SS	8.77	8.75	8.63
		PP	8.72	8.7	8.63
		SP	8.78	8.77	8.71
	Character bigrams	AC	5.17e7	13.9	9.23
		SS	1.29e9	14.23	8.61
		PP	1.82e7	13.22	9.31
		SP	4.02e8	14.97	8.77
	Character trigrams	AC	3.03e8	9.53	8.89
		SS	2.55e11	9.58	9.02
		PP	2.91e8	10.09	9.22
		SP	7.80e11	10.21	9.06
	Character unigrams (lowercased)	AC	8.61	8.59	8.49
		SS	8.67	8.65	8.56
		PP	8.48	8.47	**8.43**
		SP	8.52	8.51	8.48
	Character bigrams (lowercased)	AC	1.18e7	16.43	8.8
		SS	1.14e9	17.02	8.67
		PP	157.72	14.69	9.22
		SP	95.91	16.16	8.97
	Character trigrams (lowercased)	AC	6.55e8	9.85	9.17
		SS	1.27e11	9.93	9.4
		PP	1.96e8	10.89	9.81
		SP	2.69e10	11.24	9.45

Table 4: RMSE of leave-one-out cross-validation on the training data (default parameter settings for Lasso) with **tfidf** feature representation. Minimum values have been boldfaced. AW = all words, AC = all characters, SS = all characters except space, PP = all characters except punctuation, SP = all characters except space and punctuation.

Feature Representation	Feature Category	Feature Type	Linear Regression	Ridge Regression	Lasso
Tfidf	Word unigrams	AW	1.56e12	8.7	8.63
	Word unigrams (lowercased)	AW	1.61e12	8.72	8.66
	Character unigrams	AC	8.73	8.61	8.47
		SS	8.73	8.61	8.47
		PP	8.79	8.74	8.6
		SP	8.79	8.74	8.6
	Character bigrams	AC	3.12e10	13.2	9.51
		SS	3.69e10	15.1	9.66
		PP	2.71e10	18.91	8.87
		SP	9.82e9	19.09	8.9
	Character trigrams	AC	8.12e11	9.45	9.62
		SS	1.96e12	9.48	9.01
		PP	1.86e12	9.46	9.22
		SP	2.74e12	9.82	9.4
	Character unigrams (lowercased)	AC	8.56	8.49	**8.4**
		SS	8.56	8.49	**8.4**
		PP	8.55	8.53	8.48
		SP	8.55	8.53	8.48
	Character bigrams (lowercased)	AC	3.79e9	16.36	9.67
		SS	9.19e9	16.58	9.81
		PP	161.94	20.48	8.88
		SP	4.38e10	22.97	9.2
	Character trigrams (lowercased)	AC	2.02e12	9.56	10.15
		SS	1.86e12	9.44	8.81
		PP	7.17e11	10.28	9.93
		SP	3.98e11	10.59	9.05

We used the corresponding models on the test data as our five runs. The final results from five runs are shown in Table 5. Our Run 5 achieved the best RMSE on extroversion (8.60) and competitive results on other traits. We believe that with more parameter tuning and feature engineering (e.g., word n-grams), we can beat the performance of our existing system and be able to advance the state-of-the-art in this challenging and interesting task.

5. CONCLUSION

In this paper, we reported the design, feature engineering, and evaluation of the BESUMich system submitted to the PR-SOCO Shared Task [15]. One of our runs achieved the best RMSE on extroversion, and all five runs performed competitively. We could not experiment with word n-grams due to sparsity and runtime issues, but we hope to resolve them in future work. Future research directions include a more rigorous feature engineering and parameter tuning step, along with *feature ranking* to identify which features are the most important in this task. Another interesting idea will be to explore the *learning curve* to see how much training data is sufficient to obtain reasonable RMSE values. Similarly, a *feature curve* will be able to indicate a reasonable vocabulary size for the experiments we performed. Overall, we are hopeful that our methodology, combined with the methods presented by other participants, will significantly advance future research in this domain.

6. REFERENCES

[1] S. M. Alzahrani, N. Salim, and A. Abraham. Understanding Plagiarism Linguistic Patterns, Textual Features, and Detection Methods. *Trans. Sys. Man Cyber Part C*, 42(2):133–149, Mar. 2012.

[2] 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.

[3] S. Argamon, M. Koppel, J. W. Pennebaker, and J. Schler. Automatically Profiling the Author of an Anonymous Text. *Commun. ACM*, 52(2):119–123, Feb. 2009.

[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*, MM '14, pages 1245–1246, New York, NY, USA, 2014. ACM.

[5] A. Clauset, C. R. Shalizi, and M. E. J. Newman. Power-Law Distributions in Empirical Data. *SIAM Rev.*, 51(4):661–703, Nov. 2009.

[6] P. T. Costa Jr. and R. R. McCrae. The Revised NEO Personality Inventory (NEO-PI-R). *The SAGE Handbook of Personality Theory and Assessment*, 2:179–198, 2008.

[7] H. J. Escalante, T. Solorio, and M. Montes-y Gomez. Local Histograms of Character N-grams for Authorship Attribution. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 288–298, Portland, Oregon, USA, June 2011. Association for Computational Linguistics.

[8] D. Estival, T. Gaustad, S. B. Pham, W. Radford, and B. Hutchinson. Author Profiling for English Emails. In *Proceedings of the 10th Conference of the Pacific Association for Computational Linguistics (PACLING'07)*, pages 263–272, 2007.

[9] A. Genkin, D. D. Lewis, and D. Madigan. Large-Scale

Table 5: RMSE of five submitted runs on the test data.

Run ID	Neuroticism	Extroversion	Openness	Agreeableness	Conscientiousness
1	10.69	9.00	8.58	9.38	8.89
2	10.69	9.00	8.58	9.38	8.89
3	10.53	9.05	8.43	9.32	8.88
4	10.53	9.05	8.43	9.32	8.88
5	10.83	8.60	9.06	9.66	8.77

Bayesian Logistic Regression for Text Categorization. *Technometrics*, 49(3):291–304, 2007.

[10] U. G. Mangai, S. Samanta, S. Das, and P. R. Chowdhury. A Survey of Decision Fusion and Feature Fusion Strategies for Pattern Classification. *IETE Technical Review*, 27(4):293–307, 2010.

[11] 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*, CHASE '16, pages 8–14, New York, NY, USA, 2016. ACM.

[12] 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.

[13] F. Peng, D. Schuurmans, S. Wang, and V. Keselj. Language Independent Authorship Attribution using Character Level Language Models. In *Proceedings of the Tenth Conference on European Chapter of the Association for Computational Linguistics - Volume 1*, EACL '03, pages 267–274, Stroudsburg, PA, USA, 2003. Association for Computational Linguistics.

[14] F. Rangel, F. Celli, P. Rosso, M. Potthast, B. Stein, and W. Daelemans. Overview of the 3rd Author Profiling Task at PAN 2015. 2015.

[15] 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.

[16] F. Rangel, P. Rosso, M. Koppel, E. Stamatatos, and G. Inches. Overview of the Author Profiling Task at PAN 2013. In *CLEF Conference on Multilingual and Multimodal Information Access Evaluation*, pages 352–365. CELCT, 2013.

[17] U. Sapkota, S. Bethard, M. Montes, and T. Solorio. Not All Character N-grams Are Created Equal: A Study in Authorship Attribution. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 93–102, Denver, Colorado, May–June 2015. Association for Computational Linguistics.