Predicting Personality Traits of Authors from Text

CHIRAYU DESAI, Northeastern University, desai.ch@husky.neu.edu
AKHILESH HEGDE, Northeastern University, hegde.ak@husky.neu.edu
YUZHOU YIN, Northeastern University, yin.yuz@husky.neu.edu

This paper delineates the techniques in machine learning we used to model natural language features extracted from labelled essay extracts, for the purpose of predicting personality traits of the author of the text. The challenges encountered in this problem stem from the fact that the prediction is a task of mapping language features that do not correlate to the predicted labels. In other words, there is no quantitative feature that strictly correlates to each personality trait. To work around this, we explore numerous techniques to get results comparable to prior research conducted in this domain.

KEYWORDS

Personality Attribution, Natural Language Processing, Machine Learning, Naïve Bayes, Support Vector Machines, Neural Networks.

ACM Reference format:

Ben Trovato, G.K.M. Tobin, Lars Thørväld, Lawrence P. Leipuner, Sean Fogarty, Charles Palmer, John Smith, and Julius P. Kumquat. 1997. SIG Paper in word Format. *ACM J. Comput. Cult. Herit.* 9, 4, Article 39 (March 2010),4 pages.
DOI: 10.1145/1234

1 INTRODUCTION

Personality is a concept that relates to the nature of humans. It is an innate state of an individual that identifies him as a person possessing some traits characteristic of that personality type. Personality is more a qualitative metric than a quantitative one. As a result, it is hard to quantify the personality of a person and analyze which personalities are more or less favorable. It is also a matter of perception by people who view other people according to their biases and preconceptions of what constitutes a personality type. Due to the general nature of personality traits, we rely on a formalized standard to classify the traits known as the **Five Factor Model (FFM)** or in general terms, as the **Big Five personality traits**. This formalism of personality traits highlights the following five as the most expansive - **Extraversion, Conscientiousness, Neuroticism, Agreeableness, Openness**. The characteristics that comprise these traits are comprehensive enough to classify any person under at least one of these traits. In most cases however, a person possesses a mix of two or more of these personality traits. Our work involves the modeling of language features, as observed from essays authored by regular people, to be able to predict personality traits from solely observing features extracted from that text.

It stands to reason that the predictions we make is not entirely reflective of the nature of a person, primarily because we use features extracted from a single source, which is the extract of an essay written by that person. What we actually want to do is to identify the features of the text, which best correlates to each personality trait. Building a system capable of learning the subtleties involved in such a task, which even humans aren't adept at is representative of the significance of language and linguistic features in learning systems and can be expanded to other applications of similar nature.

2 RELATED WORK

Personality analysis is a rich field for natural language text and speech analysis. In the late 90s some work on personality detection from plain text was done by [1] .They collected stream-of consciousness essays labeled by the Big Five personality traits. They used Linguistic Inquiry and Word Count (LIWC) features. The correlation between LIWC, Thematic Apperception Test (TAT) and Personality Research Form (PRF) measures was determined. They further correlated the results with the essay and the big five personality traits. François Mairesse and colleagues build statistical models using features based on utterance and prosody, to achieve better results. They also tested the trained models on the outputs of unseen individuals [2].

In 2006 Richard B. Slatcher and colleagues examined the personalities and psychological states of the 2004 candidates for U.S. president and vice president through their use of words. They collected samples of speeches delivered by them and used LIWC to classify words. After processing the speeches with LIWC, six linguistic measures were created for analysis based on cognitive complexity, femininity, depression, age, presidentiality, and honesty [3].

Deep learning methods for this task usually involves a multilayer neural network with appropriate activation functions for feature manipulation and classification. This is used to identify nonlinear features, if any, as well as relevant features which identify the nuanced differences in features for different personality traits. Research work conducted in [4] relies on training using feature vectors constructed from the word level up and using hidden layers in the neural network to construct document vectors. Accuracy in results relies on the right feature vector extraction, filtering the corpus and identifying the features generated at each hidden layer that capture the distinctive differences between personality traits, which are eventually put together by a classifier layer.

3 DATASET

The dataset we use is a stream-of-consciousness essays, which consists of essays that are labelled with the personality traits that are most reflective of the content.

The essays talk about certain life events of the authors from their points of view. There were 2468 essays of this kind in the data set with their corresponding labels: Extraversion, Neuroticism, Agreeableness, Conscientiousness, and Openness. One essay has been removed as it had little text. This stream-of-consciousness essays dataset is available at the following link https://github.com/SenticNet/personality-detection/raw/master/essays.csv.

For our experiments we have divided the dataset into a ratio of 1:3 i.e. 75% of the dataset is used for training and 25% for testing the accuracy of prediction by our models.

4 METHODOLOGY

4.1 Feature Extraction

We extracted features of different kinds from these essays and a subset of these features were used in each model that we used to build a classifier. The features and their significance with respect to the nature the classification we're doing are elaborated below.

(1) **Word2Vec Embedding's**: We generated word2vec embedding's for each essay by averaging the embedding's over all sentences of the essay. The idea behind averaging the vectors is to condense a collection of embedding's and generate standard sized features for every essay which can be used as a semantically representative feature of the essay.

(2) **Bag of Words**: Another feature that is semantically representative of the content of each essay is the bag of words features.

(3) **TF-IDF**: The feature obtained by taking the product of the term frequency and inverse document frequency for each essay. This feature helps associate the subsets of the vocabulary that are unique to each essay which can help with prediction for essays that consist of similar patterns of word occurrence.

(4) **Sentiment Annotation**: We used the Stanford full parser to annotate every sentence of each essay with the respective sentiment annotation - which is a probability distribution over varying degrees of sentiment polarity of the sentence being annotated. The five degrees of polarity for which it generates the probability distribution are - Extremely positive, moderately positive, neutral, moderately negative, and extremely negative. We average the probability vectors over the entire essay and consider that to be a feature which captures the overall sentiment of the essay. This feature yields high correlation with the personality traits.

(5) **Stemmed Words**: We consider the bag of stemmed words of each essay as another feature which captures the most informative words and identifies occurrences of a word in different forms as the occurrence of the same word.

(6) **Quantitative Metrics**: We used a few quantitative metrics as features to make up the numbers of candidate features that may be relevant to the models we are trying to build.
- Average sentence length of the essay.
- Length of the essay.

Our intention behind generating features that may or may not be relevant to a machine learning model that we experiment with is that we perform automatic feature extraction using the K-best feature selection model, using the χ^2 distribution to fetch the K best features for a certain model. We select K by a method of trial and error, which maximizes the accuracy scores for prediction.

We observed that the word2vec and sentimental annotation features average out over the essay and don't have distinguishing impact in predicting labels. Stemming the words had a negative impact on accuracy. Consequently we have achieved our final results primarily from one or more features listed above excluding word2vec, sentimental annotation and stemming.

4.2 Modeling

(1) **Support Vector Machine**: SVMs are popular linear classifiers that essentially splits the data plotted on a multidimensional graph with a hyperplane. Ideally, the features we use in this classifier must correlate linearly with the all the classes in a way that the classifier linearly segments the space to include all records with the same trait labels. We train a single-label One *Vs* All classifier for each label, which uses the subset of features that most linearly correlate to the classes they are labelled with.

(2) **Naïve Bayes**: Naïve Bayes works fairly well for text based classification problems. The classifier makes predictions based on the learning of distribution of posterior probability. As a result, the classifier performs well even with a large dimension training set. We train the Naïve Bayes model with a subset of features listed in the previous section. It turns out that the Naïve Bayes classifier yields the best result for one of the personality traits. However, the Naïve Bayes model does not generalize well if test documents are vastly different from the training documents.

(3) **Random Forest**: Random forest is a decision tree based model, it predicts the result by averaging several results from multiple pre-built decision trees. Each decision tree has different structures learnt from the training input. In our project, multiple features are in use, but we have do not know how to weight the features to get the best results for the classification. So we experimented with the Random Forest model with the training set containing the features we choose to use and use the rest of the labelled data for validation.

(4) **Multilayer Perceptron**: Multilayer perceptron yields the best result for 4 out of the 5 personality traits classification among all models we experimented with. The reason is that it is quite flexible to work with non-linear features and also highly configurable by means of the hyper parameters. We train the multilayer perceptron with a subset of our feature set and optimize the model parameter with grid search. It turns out that with sufficient training set, the multilayer perceptron can approximate the non-linear function pretty well due to the hidden layer in its structure. However, when we increase the dimensionality of the input by including more features, the performance of the model does not increase commensurately due to the fact that Multilayer perceptron is sensitive to the scale of the dataset and it converges at the local optimum instead of the global optimum.

4.3 Hyper-Parameter Tuning

Hyper-parameters are parameters that are not directly learnt within model estimators. They are generally passed as arguments to the constructor of the estimator models classes. We have used the following concepts to tune them:

- **K-fold Cross Validation**: It divides all the samples in k groups of samples, called folds (if k = n, this is equivalent to the Leave One Out strategy), of equal sizes (if possible). The prediction function is learned using k - 1 folds, and the fold left out is used for test.

- **Grid Search Cross Validation**: It exhaustively considers all combinations of hyper-parameters, as well as uses k-fold cross validation to guess the best possible values.

For selecting hyper-parameters we used the following approach:

1. Extracted a high scoring subset of valid hyper-parameter values using K-fold and Grid Search Cross Validation. The scoring criteria used were accuracy, precision, recall and f1 score.
2. Iteratively ran our models on each combination of values to determine the best combination.

5 EXPERIMENTS

5.1 Baseline Model

For the baseline model, we used a semantic vector representation of the essays as the feature, which was used to train a multi-label classifier. The following were the parameters of the model:

(1) **Model**: Neural Network classifier
(2) **Primary feature**: word2vec averaged over all sentences of an essay (Embedding size = 50)

We found the F1 score for the predictions of this model to be roughly 50%, which is considerably worse than the best results achieved in related work.

5.2 Naïve Bayes

Naive Bayes, as explained in previous sections, is widely used in applications of text classification problems. We expected to achieve fairly good results from an implementation of this model.

(1) **Model**: Multinomial Naive Bayes
(2) **Primary feature**: TF-IDF on lowercase documents

The multinomial Naive Bayes classifier surpassed the best results previously seen for one of the personality traits for which the tone of the essays was predominantly linear. We tuned the smoothening parameter alpha by methodology elaborated in section 4.3.

5.3 Multilayer Perceptron

Multilayer perceptron is a good choice for most classification systems. We wanted a model that can adapt to non-linear features as well.

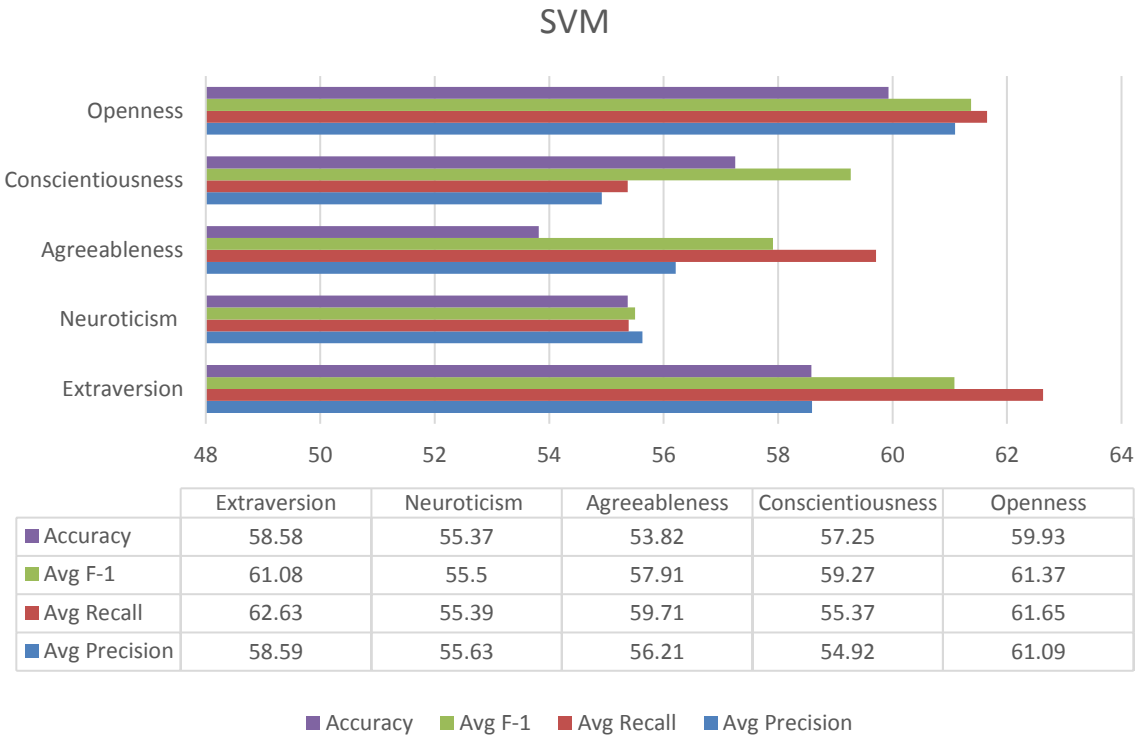
(1) **Model**: Multilayer Perceptron
(2) **Primary feature**: TF-IDF and/or Sentiment probabilities for five degrees of sentiment polarity.

We found the accuracy of predictions for this model to be the best for three of the five personality traits. We did a fair bit of hyper-parameter tuning and feature selection for these models in order to achieve proper accuracy. The results depicted are the average results over 10 consecutive runs of the model.

6. EVALUATION RESULTS

We have tuned our classifier models Hyper-parameters by cross validation and evaluated our results by comparing it against labelled test set, which is a 25% subset of the Stream of consciousness essays dataset. The results for the model experiments are summarized below. Some general observations and details about the results are listed below.

(1) Support Vector Machines do not generalize well beyond the most trivial linear features. So, for any subset of our set of features, the classifier works by identifying relatively linearly correlating features to do the classification among all the possible label predictions.

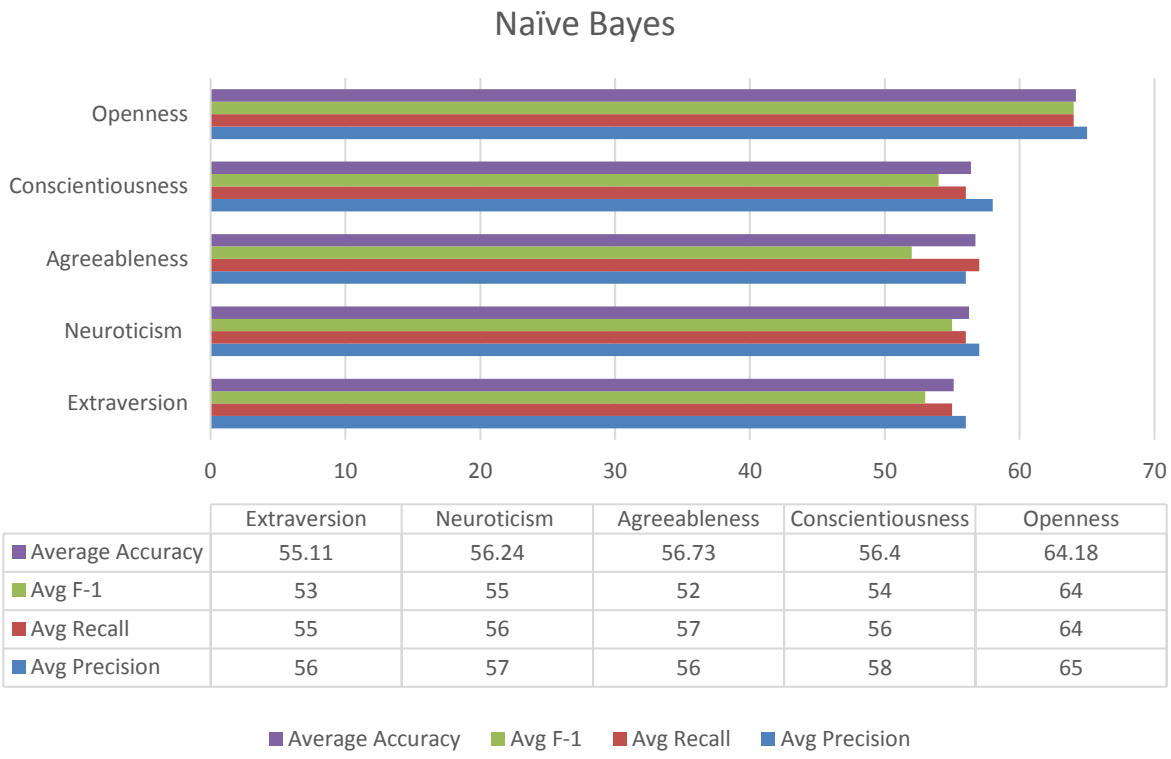


	Extraversion	Neuroticism	Agreeableness	Conscientiousness	Openness
Accuracy	58.58	55.37	53.82	57.25	59.93
Avg F-1	61.08	55.5	57.91	59.27	61.37
Avg Recall	62.63	55.39	59.71	55.37	61.65
Avg Precision	58.59	55.63	56.21	54.92	61.09

Figure 1: Evaluation results for SVM

(2) Random Forests cannot tune the weights of the model to improve classification accuracy because it weighs existing features to get the best possible accuracy. The training method for the model is, as the name suggests - 'random', and the features are randomly weighted to try to achieve the best classification accuracy. The model has no specific direction for optimizing its weights and hence, we were not able to achieve results that were any better than those of SVM.

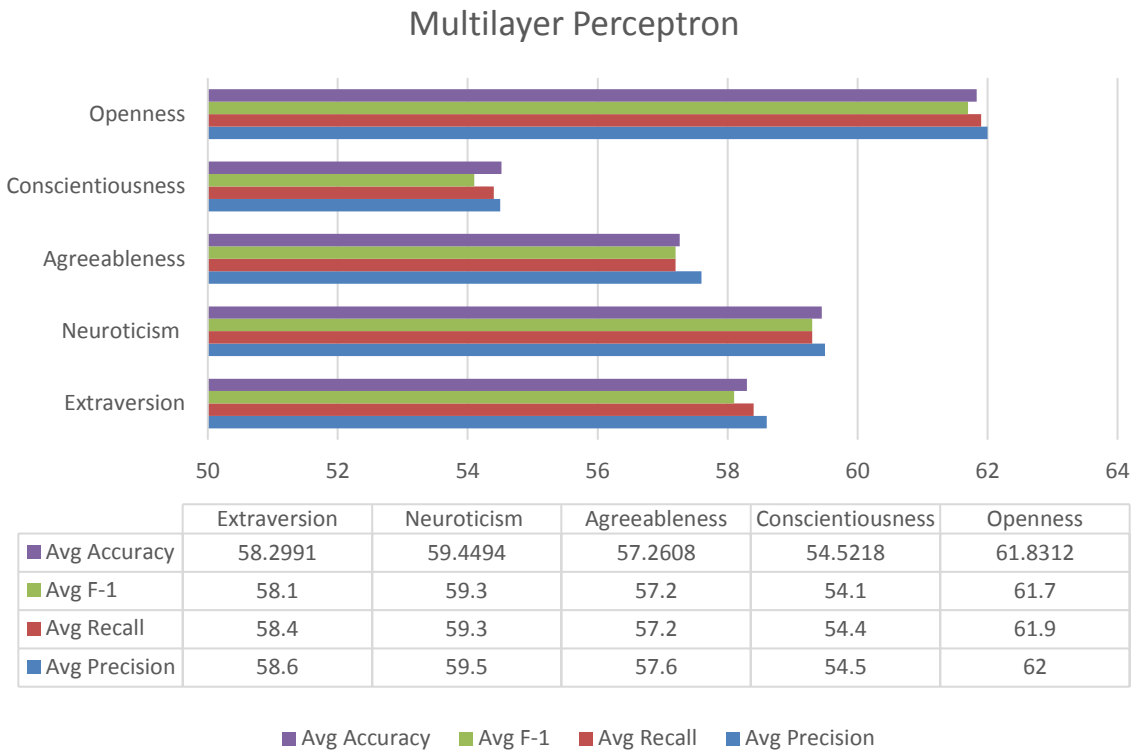
(3) Naive Bayes gave us the best classification accuracy for one out of the five personality traits, which is 'Openness'. We think this happens because the trait for Openness is generally associated with a lexicon that is indicative of characteristics that relate to Openness. Since Naive Bayes operates on prior probabilities of the words in the training set, it is able to classify test data possessing this trait more than any other.



	Extraversion	Neuroticism	Agreeableness	Conscientiousness	Openness
Average Accuracy	55.11	56.24	56.73	56.4	64.18
Avg F-1	53	55	52	54	64
Avg Recall	55	56	57	56	64
Avg Precision	56	57	56	58	65

Figure 2: Evaluation Results for Naïve Bayes Classifier

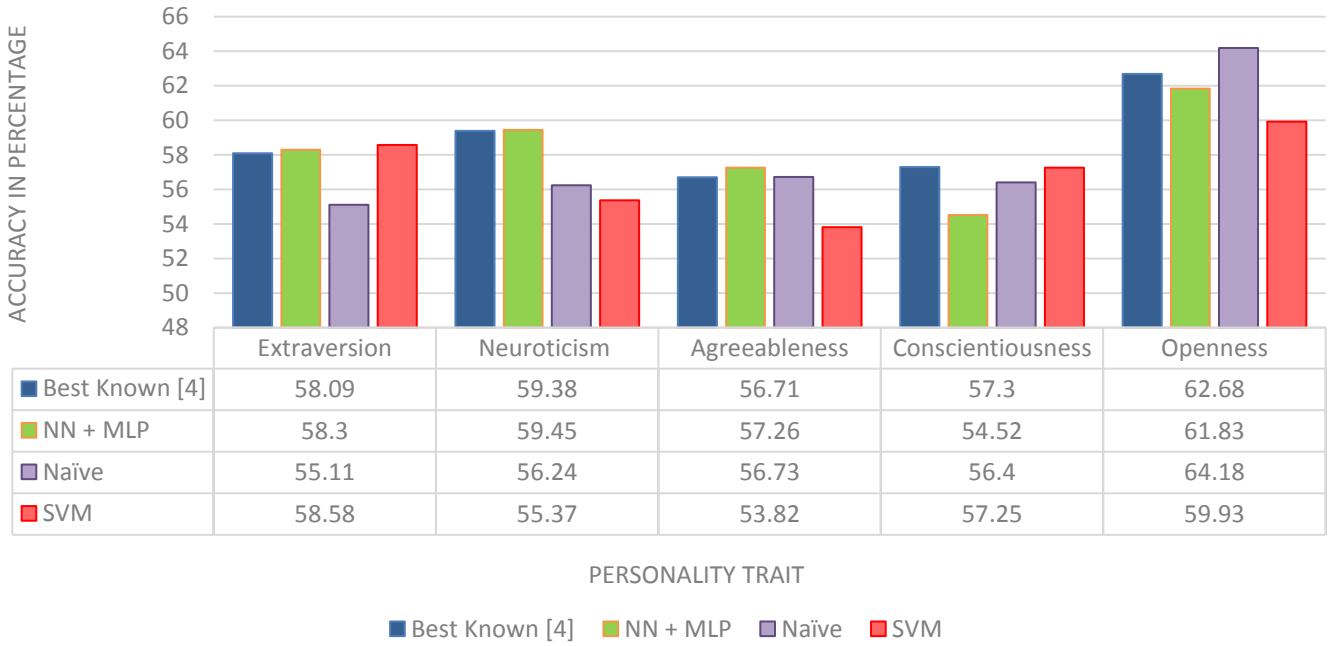
(4) Multilayer Perceptron is generally the best model for this task. Among all the models we experimented with, this gave us the best results, primarily because of the flexibility of parameter tuning for the model. We ran grid search to tune the hyper parameters of the model which helped to improve the classifier.



	Extraversion	Neuroticism	Agreeableness	Conscientiousness	Openness
Avg Accuracy	58.2991	59.4494	57.2608	54.5218	61.8312
Avg F-1	58.1	59.3	57.2	54.1	61.7
Avg Recall	58.4	59.3	57.2	54.4	61.9
Avg Precision	58.6	59.5	57.6	54.5	62

Figure 3: Evaluation Results for Multilayer Perceptron Classifier

The figure below summarizes the results in comparison with the best known results for this study according to the work done by Majumder and peers in 2017.



	Extraversion	Neuroticism	Agreeableness	Conscientiousness	Openness
Best Known [4]	58.09	59.38	56.71	57.3	62.68
NN + MLP	58.3	59.45	57.26	54.52	61.83
Naïve	55.11	56.24	56.73	56.4	64.18
SVM	58.58	55.37	53.82	57.25	59.93

Figure 4: Comparative Analysis of Results

7. CONCLUSION

Prediction of personality traits from text is a non-trivial task. Identifying what characteristics constitute each personality type by means of using learning systems still requires the experimentation with new techniques to get better prediction accuracy. Currently, our models are performing as expected. As we read through the publications that resulted from previous work in this space, the best accuracy achieved is around 50-60%. We are considering a baseline of 55% accuracy, with our baseline model, which uses all the words in the essay documents to model the classifier.

We then proceeded to model a subset of ten features we gathered, for each personality trait, by doing feature selection. Doing single label classification with automatic feature selection let us use the most salient features for each trait. From the results we obtained, we can say that a single model with unchanging parameters cannot yield the best results. That is why we did single label classification for each personality trait, the model parameters for which we obtained by means of hyper parameter tuning using grid search.

8. FUTURE WORK

Future directions for this project can involve using full parses of the essay documents to leverage similarities in substructures of the parse trees. We think this can help with identifying the tone and sentence structure that are common to the writing styles of people of the same personality types.

9. ACKNOWLEDGMENTS

This project wouldn't have been possible without the constant assistance and guidance from Professor Lu Wang and the course teaching assistant Liwen Hou.

REFERENCES

[1] Pennebaker, J. W., King, L. A. Linguistic styles: Language use as an individual difference. (1999)
[2] Mairesse F., Walker M., Mehl M., Moore R. Using Linguistic Cues for the Automatic Recognition of Personality in Conversation and Text. (2007)
[3] Bush, G.H., Chung, C.K., Edwards, J.G., Pennebaker, J.W., Slatcher, R.B., Stone, L.D. Winning words: Individual differences in linguistic style among U.S. presidential and vice presidential candidates. (2006)
[4] Majumder N., Poria S., Gelbukh A. and Cambria E Deep Learning-Based Document Modeling for Personality Detection from Text. (2017)
[5] Implementation of a hierarchical CNN based model to detect Big Five personality traits. https://github.com/SenticNet/personality-detection/raw/master/essays.csv (2017)