Personality Detection Based on Deep Learning

Yang Cai, Bowen Yin, Huan Tan

1 Project Description

Personality is a fascinating topic among all time. Personality is a combination of a person's emotion, idea, thought and even life credo. Personality also affects how we interact with others, make decisions and deal with troubles. While this being so important, we often have little information about personality. People may hardly identify themselves personality clearly.

In 1884, Sir Francis Galton firstly came up with the idea that personality could be decomposed into different sub-features (PE and ST,). As time goes on, this idea got to be adapted and more completed. Then the theory of big five personality traits came out. Big five personality traits suggest that humans personality can be described by five primary factors: extroversion, conscientiousness, agreeableness, neuroticism, and openness. Extroversion indicates how outgoing a person is. Conscientiousness describes how organized and efficient the person could be. Agreeableness represents the tendency of being cooperative versus suspicious of a person. neuroticism describes the level of sensitivity a person is. And openness reflects the degree of curiosity and creativity a person would have. Each of the five factors represents a unique aspect of human personality.

In spoken language analysis, an utterance is the smallest unit of speech. It is well known that utterances convey a great deal of information about the speaker. Sentiment analysis is a huge category under natural language processing, since text data often carries a great amount of information about the person's emotion. However, personality identification seems to be a less common field than other ones. With the importance of personality analysis, we think it would be interesting to do text classification based on big five personality traits using essays written by corresponding persons, and analyze how different writings or words used could affect human personality.

Upon project, we presented natural language processing methods to detect a person's personality traits based on essays. We also quantified the connections of writings on personality. Different methods for extracting data features were used, including convolutional neural network building word embeddings, features from other paper, as well as TF-IDF(term frequency inverse ducument frequency). Thus, instead of having essay text data for each person, we ended up with numeric data representing word meanings and word counts used per essay. Later on, data was splitted into training set and test set. Upon training set, different machine learning algorithm were performed for binary classification, predicting person's personality.

2 Related Work

With broad applications, people are becoming more and more interested in the research field of personality detection. Some early works on automated personality detection from plain texts use traditional machine learning methods and feature engineerings. The most common ones are Support Vector Machine(SVM) and Linguistic Inquiry and Word Count (LIWC) approaches. For example, SVM based methods are used along with manipulated lexical and grammatical features for classification by Scott

Introvert	Extravert
Well, right now I just woke up from a mid-day nap. It's sort of weird, but ever since I moved to Texas, I have had problems concentrating on things. I remember starting my homework in 10th grade as soon as the clock struck 4 and not stopping until it was done. Of course it was easier, but I still did it.	ahh. gotta love that screech the chairs let when you push them back. ahhhhh. well. oh yeah at the meeting I met several people. Caleb seems rather worried about one of the women. though he is bound and like wise I am unable to speak ill of her. Well I am in charge of running our booth Monday.
Unconscious	**Conscious**
Well, here we go with the stream of consciousness essay. I used to do things like this in high school sometimes. They were pretty interesting, but I often find myself with a lack of things to say. I normally consider myself someone who gets straight to the point.	I can't believe it! It's really happening! My pulse is racing like mad. So this is what it's like. now I finally know what it feels like. just a few more steps. I wonder if he is going to get any sleep tonight!? I sure won't! Well, of course I have a million deadlines to meet tomorrow so I'll be up late anyway. But OH!

Table 1: Essays labeled by Big Five personalities

Nowson (Nowson and Oberlander, 2006). One common problem for dealing text data is that corpora labeled data are usually sparse, and researchers have used some approaches and showed great work in building non-sparse novel feature set. Some of those popular syntactic features utilization include POS tags for dependency relations between tokens, and latent semantic analysis for topic modelings.

LIWC is another popular method used to generate text features. In the paper "Using linguistic cues for the automatic recognition of personality in conversation and text", LIWC features are combined with other features to do prediction on personality (Mairesse et al., 2007).

Deep learning has its' increasing popularity in NLP field, as well as personality detection. The model described in Kalghatgi's paper presents a neural network based method to do personality prediction of users(Kalghatgi et al., 2015). In this model, a fundamental multilayer perceptron (MLP) neural network takes an input of hand-crafted grammatical and social behavioral features from each user, and then it assigns a label to each of the 5 personality traits. In 2016, Ming-Hsiang and his colleagues build a Recurrent Neural Network (RNN) based system, exploiting the turn-taking of conversation for personality detection (Su et al., 2016). In the work, RNNs takes LIWC-based and grammatical features as inputs, and are employed to model the temporal evolution of dialog.

Under the assumption that character sequences are syntactically and semantically informative to the words, a character level word representation model has been proposed by Wang Ling in 2015 (Ling et al., 2015). Based on the long short-term memory network (LSTM), the model learns the embeddings of characters and how they can be used to construct words. Topped by the softmax activation function at each word, the model is applied to the tasks of language modeling and other NLP tasks.

3 Data Collection Method

Dataset used in this project is the James Pennebaker and Laura Kings stream-of-consciousness essay dataset (Pennebaker and King, 1999), which contains 2,468 anonymous writings tagged with the corresponding authors personality traits ratings in binary form: extroversion, neuroticism, agreeableness, conscientiousness, and openness. In the writings, each student were told to write whatever comes into their mind for 20 minutes, and the corresponding personality traits were tested using the Big Five Inventory questionnaires. There are total of 6 features in data, the first feature is the writing text feature, where each element represents the whole writing content(essays) of the subject, and rest of the features are the five personality scores (binary: Y/N). Data was randomly splited into 80% train set and 20% test set.

Data stream-of-consciousness essay was collected through an open-sourced website, and all personal information were de-identified before any public ac-

cess. There is no related ethical issue needing to be concerned with for this project.

4 Method Description

The purpose of this paper is to do document-level classification based on the author's Big Five personality traits. Our approach can be summarized in the following steps:

step 1 Collect individual essays

step 2 Collect ratings on authors' Big Five personality traits

step 3 Word Embedding

step 4 Extract relevant features

step 5 Apply classification models based on extracted features

step 6 Test model and do comparison

For each document, we used three different methods for feature extraction: convolutional neural network, Mairesse features, TF-IDF. After that we applied SVM, logistic regression, neural network separately on these features and did comparison with the baseline.

4.1 Data Pre-processing

Data pre-processing included sentence splitting, data cleaning, and unification. We reduced all letters to lowercase. All the punctuation marks according to the priorities should be dealt with. For Example: ".", "," ,"?", "!" are important punctuation that should be retained while others need to be removed.

- Sentence splitting
  - Each essay is split into sentences by periods, exclamation marks, and question marks.
  - Each sentence is split into tokens by white space.
- Data cleaning
  - Within each sentence, punctuation tokens were removed.
  - For numbers appeared in the writings, all are represented by the same token.
  - Words that are not purely comprised of alphabetical characters are removed.
  - Words that have length less than 2 are removed.
- Unification: Lower cases.

4.2 Word Embedding

Model represents individual words by Word Embeddings in a continuous vector space. Word2Vec for Word Embeddings was used. This gives us a variable-length feature set for the document: each document is represented by sentence vectors, and each sentence vector is represented by fixed-length word feature vectors. Word2vec "vectorizes" about , and by doing so it makes natural language computer-readable we could start to perform powerful mathematical operations on words to detect their similarities From Google pre-trained newspaper Word2Vec, each word is represented by a length 300 vector in a continuous vector space. Words that are not in pre-trained Word2Vec model are randomly assigned to length 300 vectors uniformly valuing between -0.25 to 0.25. Each word is converted to a 1-dimensional vector with length 300.

4.3 Feature Extraction

Several methods were used for feature extraction in this project, which includes self-trained document vectors by using Word2Vec and CNN, feature sets from other papers for personality detection, and TF-IDF. By including those three methods, both syntactic and semantic information would be captured in feature set.

4.3.1 Feature Extraction Using CNN

Before doing personality classification on documents, we need to map documents to a vector space. The mapping is separated into the following 4 steps:

1. Sentence vectorization: convert sequences of words in each sentence to fixed-shape matrix
2. Document vectorization: convert sequences of sentence matrices to document cubes
3. Convolutional Neural Network: use CNN to convert each document cube into an 1-dimensional vector

Sentence Level Each sentence is composed of multiple words and can be represented by a variable number of fixed-length word feature vectors. Each sentence is $W \times 300$ matrix, where w represents the maximum length of sentence. In the James Pennebaker and Laura Kings stream-of-consciousness essay dataset, maximum length of sentence equals 148.

Document Level Suppose S represents the number of sentences in a document and each document could then be represented by a 3-dimensional cube with shape $W \times 300 \times S$.

4.3.2 Convolutional Neural Network

Now the dataset is a set of documents, where each document is represented by a sequence of sentences, each sentence is represented by a sequence of words, and each word is a numeric vector of fixed length. Each document is represented by a real-valued 3-dimensional cube. In order to do classification, we need to extract features of each document from these cubes.

Input Layer We built a convolutional neural network to do feature extraction. The input layer is a document i.e. 3-dimensional real-valued array from $R_{W \times E \times S}$, in which S is the maximum number of sentences in a document across all documents, W is the maximum number of words in a sentence across all documents, and E is the length of Word2Vec Embedding. In The James Pennebaker and Laura Kings stream-of-consciousness essay dataset, E is set to be 300, W equals 148, S equals 312.

Convolution Layer In the convolution layer, we used 3 different sizes of convolutional filters to extract n-gram $(n = 1, 2, 3)$ features. The size of kernel to extract n-gram $(n = 1, 2, 3)$ feature equals $n \times 300$. Noticing that the size of input layer is $W \times E \times S$, the convolution did was within each slice with size $W \times E$. For each n, 200 filters with the same size $n \times 300$ were used.

Pooling Layer After convolution, we used Leaky Relu as the activation function and did max pooling. The reason we used Leaky Relu, is to avoid sparsity of the output layer. From three different convolutional layers along with max pooling, for each document, the unigram, bigram, and trigram features are

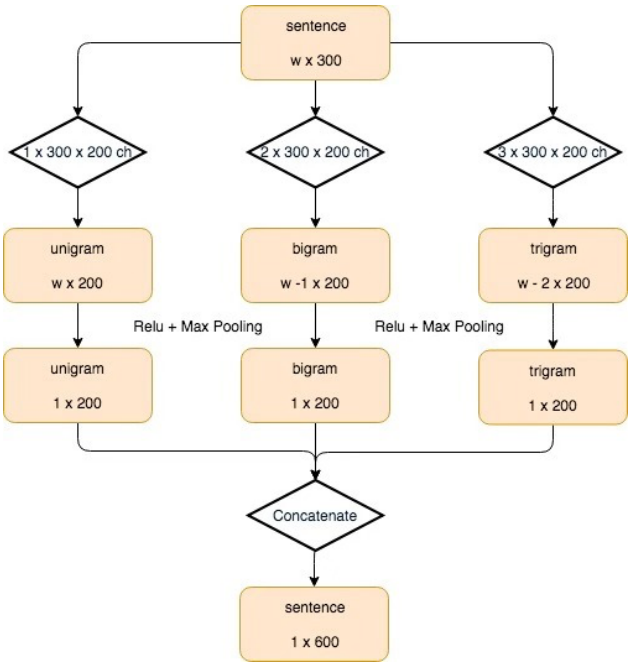

Figure 1: CNN Architecture

represented as a 1 x 200 vector. By concatenation, each document could be represented as a 1×600 vector.

Dropout Layer In order to avoid overfitting, a dropout layer with dropout rate 0.4 was added after the pooling layer.

Binary Classification After extracting n-gram features, five CNN models using TensorFlow for classification on five traits of personality were built to obtain document level vectors. By adding an extra linear layer activated by Sigmoid function, the output layer predicted the probability distribution over binary classes on all traits given the input document. The network was trained to minimize the cross-entropy of the predicted and true distributions. Weights in each convolution kernel were trained corresponding to a linguistic feature detector that learns to recognize a specific class of document.

4.3.3 TF-IDF Features

Another feature extraction method we used was TF-IDF. TF-IDF is widely used in information retrieval and its so useful a tool to extract lexical characteristics. All of unigram, bigram and trigram were considered within this feature, while the top 200 features with highest TF-IDF variances were taken.

When dealing with the corpus, a set of stopwords was removed. Ideally, high variance of a features TF-IDF score indicates that there is a large variability of token appearance between documents, which might be more representative. In terms of TF-IDF, each document is represented by a 1×200 vector.

4.3.4 Mairesse Features

Franois Mairesse developed a document-level feature set for personality detection using the Pennebaker data, which consists of 84 features(Mairesse et al., 2007). This paper generated linguistic variation projecting multiple personality traits continuously, by combining and extending previous research in statistical natural language generation.
The Mairesse's features are the combination of the Linguistic Inquiry and Word Count features, Medical Research Council features, utterance-type features, and prosodic features. Examples of the features included in this set are the word count and average number of words per sentence, as well as the total number of pronouns, past tense verbs, present tense verbs, future tense verbs, letters, phonemes, syllables, questions, and assertions in the document.

4.4 Classification

In Mairesse's paper, they applied classification models including decision tree, Nearest neighbour, Naive Bayes, JRip rule set, Adaboost and Support vector machines. In their paper, support vector machine performed best. In order to compare our personality detection techniques with that in Mairesse's paper, we applied support vector machine for classification. Besides support vector machine, we applied logistic regression, k-nearest neighbors and neural network as well. Among which support vector machine and logistic regression performed better.

4.4.1 Support Vector Machine

Support vector machine (SVM) is a supervised learning model that uses a hyperplane to separate binary classes. The goal of SVM is to maximize observations that are correctly identified, as well as the distance between hyperlane and observation points. For the cases where observations are not linearly separable, a regularization term was added to find the appropriate hyperplane. To build a binary classifier, we solved the following optimization problem:

$$min \left[\frac{1}{n} \sum_{i=1}^{n} \max\left(0, 1 - y_i(w \cdot x_i - b)\right) \right] + \lambda \|w\|^2 \qquad (1)$$

Where $\lambda\|w\|^2$ is the regularization term, and the first part of equation 1 represents the hinge loss. This equation would be equivalent as solving

$$\begin{aligned} \min_{\beta,\beta_0} \quad & \frac{1}{2}\|\beta\|^2 + C\sum_{i=1}^{N} \xi_i \qquad (2) \\ \text{s.t.} \quad & Y_i(x_i^T\beta + \beta_0) \geq 1 - \xi_i, \ \ \forall i \\ & \xi_i \geq 0, \ \ \forall i \end{aligned}$$

and got our classifier $f(x) = x^T\beta + \beta_0$.
We then used (Platt, 1999) method to provide a probabilistic output, by fitting a sigmoid function

$$p(y = 1|f) = \frac{1}{1 + \exp(Af + B)} \qquad (3)$$

The two parameters A and B were obtained by minimizing the log likelihood of the training data,

$$\min \ -\sum_i t_i \log(p_i) + (1 - t_i)\log(1 - p_i) \qquad (4)$$

where $t_i = \dfrac{y_i + 1}{2}$, $p_i = \dfrac{1}{1 + \exp(Af_i + B)}$.

4.4.2 Logistic Regression

One fairly simple way to form binary classification was to perform binary logistic regression model, which would proceed as follows:

$$\ln \frac{\Pr(Y_i = 1)}{\Pr(Y_i = 0)} = \boldsymbol{\beta} \cdot \mathbf{X}_i \qquad (1)$$

Using the fact that all probabilities must sum to one, we could find the probabilities of belonging to certain class as below:

$$\begin{aligned} \Pr(Y_i = 1) &= \frac{e^{\boldsymbol{\beta}\cdot\mathbf{X}_i}}{1 + \sum e^{\boldsymbol{\beta}\cdot\mathbf{X}_i}} \qquad (2) \\ \Pr(Y_i = 0) &= \frac{1}{1 + \sum e^{\boldsymbol{\beta}\cdot\mathbf{X}_i}} \end{aligned}$$

5 Results

In order to evaluate our model, we are going to compare the performances of different classifiers based on the test data.

Feature Set	None	CNN		CNN+Mairesse		CNN+TF-IDF		CNN+Mairesse+TF-IDF	
Classifier	Base	LR	SVM	LR	SVM	LR	SVM	LR	SVM
Extroversion	55.13	53.04	52.43	53.04	53.44	55.47	54.05	56.48	**57.69**
Conscientiousness	55.28	52.22	52.50	55.06	54.66	**58.91**	**58.30**	**58.70**	55.06
Agreeableness	55.35	52.43	51.82	54.25	53.24	**55.87**	53.85	**56.88**	54.86
Neuroticism	58.09	51.82	50.63	54.86	53.64	**59.51**	54.66	58.10	54.66
Openness	59.57	52.43	50.81	52.23	52.02	56.88	52.83	58.10	54.86

Table 2: Classification Accuracies

5.1 Evaluation

Data was randomly split into a 80% size training set and a 20% size testing set prior to any data preparation. After the models being developed using training set, we made prediction on Big Five personalities traits based on essays in the testing set. Then the accuracy of predictions on each trait was calculated, and then we compared how each model performed within each trait.

Many papers have done similar researches on the same data. Baseline metrics chosen for this project are results from paper "Using Linguistic Cues for the Automatic Recognition of Personality in Conversation and Text" in 2007(Mairesse et al., 2007). As mentioned above, in Mairesse's paper, they applied classification models including decision tree, Nearest neighbour, Naive Bayes, JRip rule set, Adaboost and Support vector machines. We chose the highest accuracy across all classification methods in Mairesse's paper as our baseline. In order to compare compare personality detection techniques with that in Mairesse's paper, we applied similar classification methods to the same dataset and see if our features performs better.

5.2 Experiment Results

CNN features were extracted by using five independent networks for five different personalities. Within each model, we used Stochastic Gradient Descent with step size 0.1 to train it. In our experiment, all the networks converged in less than 100 epochs.

Table 2 presented the classification results for different personality traits. Overall, TF-IDF features improved the model accuracy. Different classifiers were attempted, however, neural networks and random forest did not perform as good as SVM and logistic regression. Something interesting was that adding more features did not always increase accuracy correspondingly. For predicting Neuroticism personality, logistic regression with CNN and TF-IDF (59.51) performed better than logistic regression with CNN, Mairesse, and TF-IDF feature set (58.10). In this case, the extra Mairesse feature set did not carry more information than what's known already. As for Openness personality trait, our model did not beat baseline (59.57). The closest one we had was from logistic regression using CNN, Mairesse and TF-IDF feature set (58.10).

6 Conclusion

In conclusion, personality detection is a fascinating field with lots of unknowns. With this project, by using more features and different models, our model accuracy surpassed baseline accuracy on Extroversion, Conscientiousness, Agreeableness and Neuroticism personality traits. For classifiers, logistic regression worked the best among all. Support Vector Machine performed well as well. Other methods such as two layers neural networks were tried but didn't work. As for feature set, CNN trained features were used as base features. Based on CNN features, adding Mairesse features increased model accuracy slightly, and adding TF-IDF features increased model accuracy more significantly. However, more features is not always better. Sometimes using feature set with CNN and TF-IDF had better performance than CNN, TF-IDF and Mairesse feature set.

In the future, we plan to apply other models on Openness personality trait to see how we could beat the baseline accuracy.

7 Individual Contributions

Yang Cai: CNN features, Mairesse features
Bowen Yin: TF-IDF features
Huan Tan: Classification

References

Mayuri Pundlik Kalghatgi, Manjula Ramannavar, and Nandini S Sidnal. 2015. A neural network approach to personality prediction based on the big-five model. *International Journal of Innovative Research in Advanced Engineering (IJIRAE)*, 2(8):56–63.

Wang Ling, Tiago Luís, Luís Marujo, Ramón Fernandez Astudillo, Silvio Amir, Chris Dyer, Alan W Black, and Isabel Trancoso. 2015. Finding function in form: Compositional character models for open vocabulary word representation. *arXiv preprint arXiv:1508.02096*.

François Mairesse, Marilyn A Walker, Matthias R Mehl, and Roger K Moore. 2007. Using linguistic cues for the automatic recognition of personality in conversation and text. *Journal of artificial intelligence research*, 30:457–500.

Scott Nowson and Jon Oberlander. 2006. The identity of bloggers: Openness and gender in personal weblogs. In *AAAI spring symposium: Computational approaches to analyzing weblogs*, pages 163–167. Palo Alto, CA.

Shrout PE and Fiske ST. Personality research, methods,and theory. *Psychology Press*.

James W. Pennebaker and Laura A. King. 1999. Linguistic styles: Language use as an individual difference. *ournal of personality and social psychology*, 77(6):1296–1312.

John C. Platt. 1999. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In *ADVANCES IN LARGE MARGIN CLASSIFIERS*, pages 61–74. MIT Press.

Ming-Hsiang Su, Chung-Hsien Wu, and Yu-Ting Zheng. 2016. Exploiting turn-taking temporal evolution for personality trait perception in dyadic conversations. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 24(4):733–744.