Personality Detection from Text Based on Big Five Model

Peng Zhou (pzhou10@ucsc.edu) , Yidan Weng (yweng5@ucsc.edu) , Mahdi Rahimi (marahimi@ucsc.edu)

Personality is the combination of characteristics or qualities that form an individual's distinctive character. We have our own special personality traits, which would affect our whole life, every choice you made, every friend you make, what kind of book you read or movie you watch and so on, also make a role in how other people see you, and how they make every choice related to you.

There are many different theories about personality, but the most popular theory in psychology academia is called Big Five [1], also known as OCEAN. It has five traits to measure personality which are Openness, Conscientiousness, Extraversion, Agreeableness, and Neuroticism.

Texts often reflect various aspects of the author's personality. If we can detect the 'Big Five' personality of authors based on the text they write using machine learning, it is really exciting! Based on the paper we have found, we are going to work on our final project.

We would like to implement a Big Five personality traits detection method [2] which introduces a document modeling technique to extract personality traits using neural networks. The paper presenting the method uses Convolutional Neural Networks for feature extraction and document modeling. Nevertheless, we would like to build our own document modeling using Recurrent Neural Networks. If time permits, we would implement the paper's CNN modeling as well and compare the results. We will use James Pennebaker and Laura King's stream-of-consciousness essay dataset [3] which contains 2,468 anonymous essays tagged with the authors' personality traits: EXT, NEU, AGR, CON, and OPN. We will use this data set for training and test as well. Stochastic Gradient Descent will be used to train the networks. We will build and train five separate neural networks with the same architecture which will be used to detect each of the five personality traits. The steps we would like to take are as follows:

- *Preprocessing:* This step includes tasks such as sentence splitting and reduction to lower case. There are also sentences that are missing periods at their end and this causes absurdly long sentences. We will split such sentences to sentences of 20 words.

- *Filtering:* We will remove sentences that do not carry information about personality traits. In order to do so, we remove sentences that do not include any emotionally charged words. In order to detect emotionally charged words, we use NRC Emotion Lexicon [4].

- *Feature extraction:*

1. Using Recurrent Neural Networks (our approach):

First, we do word vectorization by using fixed-length *word2vec* word embeddings. After that, we do document vectorization by using an LSTM unit which accepts variable-length sequence of words and generates a fixed-length document feature vector. However, this approach might not give an optimal result. Therefore, we would like to use an LSTM layer first to do sentence vectorization. In other words, it will get a variable-length sentence and generates fixed-length sentence feature vector. Now a document would be a variable-length sequence of sentence vectors. Hence, we will have another LSTM layer that gets such sequences and generates a fixed-length document feature vector.

2. Using Convolutional Neural Networks (the paper's approach):

We use a deep CNN. Its initial layers process the text in a hierarchical manner. First, we do word vectorization by using fixed-length *word2vec* word embeddings. Hence, sentences are represented as a variable number of word vectors.

Next, we do sentence vectorization: we use three convolutional filters to extract unigram, bigram, and trigram features from each sentence. After max pooling, the sentence vector is a concatenation of the feature vectors obtained from these three convolutional filters. This sentence vector is fixed-length. At this level, documents are represented as a variable number of such fixed-length sentence embeddings.

Finally, we do document vectorization: the variable-length document vector is reduced to a fixed-length document vector. This process is as follows: the document vector is a variable-sized concatenation of all its sentence vectors. We assume that the document has some feature if at least one of its sentences has this feature. Each sentence is represented as a n-dimensional vector. To obtain the document vector, for each of these n features, we take the maximum across all the sentences of the document. This gives a n-dimensional real-valued vector of the whole document.

- *Extra document-level feature extraction:* In addition to the above feature extraction, we will use Mairesse baseline feature set [5] and apply it on the texts to extract global features such as word count and average sentence length. This feature vector is finally concatenated to the previous feature vector to form the final feature vector.

- *Classification:* We will use a fully connected neural network with one hidden layer to have a binary classifier (Yes/No) for the personality trait. Please remind that we will have 5 networks trained for each of the 5 personality traits.

Implementation Platform:

As for the implementation platform, we are currently thinking about implementing it either with Keras or Tensorflow.

References:

1. J. Digman, "Personality Structure: Emergence of the Five-Factor Model," Ann. Rev. Psychology, vol. 41, 1990, pp. 417–440.

2. N. Majumder, S. Poria, A. Gelbukh, and E. Cambria. Deep learning-based document modeling for personality detection from text. IEEE Intelligent Systems, 32(2):74–79, 2017.

3. J.W. Pennebaker and L.A. King, "Linguistic Styles: Language Use as an Individual Difference," J. Personality and Social Psychology, vol. 77, no. 6, 1999, pp. 1296–1312.

4. http://saifmohammad.com/WebPages/NRC-Emotion-Lexicon.htm

5. F. Mairesse et al., "Using Linguistic Cues for the Automatic Recognition of Personality in Conversation and Text," J. Artificial Intelligence Research, vol. 30, 2007, pp. 457–500.